AN

INTERNSHIP REPORT

ON

SMART HEALTHCARE PREDICTION MANAGEMENT SYSTEM

PROJECT

BY

KAMAL ACHARYA

(Tribhuvan University)

Date: 2025/06/07

ABSTRACT

The purpose of the project entitled as “Smart Health Prediction” is to develop software which is user friendly simple, fast, and cost – effective. It deals with the collection of patient’s information, diagnosis details, etc. Traditionally, it was done manually The main function of the system is register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully System input contains patient details, diagnosis details, while system output is to get these details on to the screen. The Smart Health Prediction can be entered using a username and password. It is accessible either by an administrator. Only they can add data into the database. The data can be retrieved easily.The data are well protected for personal use and makes the data processing very fast.

This is the document of the Project proposal for developing a health care system. This system is fully updated and we had provide some new things in it. The project aimed to build a fully functional system in order to achieve the efficiency in faster heath treatment and online consultation system. The overall mission of system development is to make the primary treatment quickly and easily complete the Online Consultation System.

We propose a system that allows users to get instant guidance on their health issues through an intelligent health care system online. The system is fed with various symptoms and the disease/illness associated with those systems. The system allows user to share their symptoms and issues. It then processes user's symptoms to check for various illnesses that could be associated with it. Here we use some intelligent data mining techniques to guess the most accurate illness that could be associated with patient's symptoms.

This Project “Smart Health Prediction” aims at maintaining patient health records and even getting appointments from various doctors for related treatments. The system user must register as a member of this system and keep updating his medical history. Patients then can select from a list of specialized doctors for respective treatments such as (skin specialist, ENT specialist, cardiologist, dietitian etc.)At particular locations. Patients may also select suitable appointment timings for their meeting. Patients can also order their medicines and perform online transactions.

INDEX

S.NO	CONTENTS	PAGE NO
1.	**INTRODUCTION** 1.1 BACKGROUND 1.2 PROBLEM STATEMENT 1.3 PROPOSED SYSTEM	1 – 9
2.	**SYSTEM ANALYSIS**	
3.	**REQUIREMENT ANALYSIS** 2.1 HARDWARE REQUIREMENTS 2.2 SOFTWARE REQUIREMENTS 2.3 INTRODUCTION TO SOFTWARE ENVIRONMENT	10 – 16
4.	**SYSTEM DESIGN** 4.1 UML DIAGRAM 1.Usecase Diagram 2.Dataflow Diagram 3.Sequence Diagram	28 – 72
5.	**IMPLEMENTATION** 5.1 IMPLEMENTATION 5.2 CODE	
6.	**SCREENSHOTS**	
7.	**SYSTEM TESTING** 5.1.TYPES OF TESTS 1.Unit testing 2.Integration testing 3.Functional test 5.2.SYSTEM TEST 1.White Box Testing 2.Black Box Testing	72 – 77
8.	**CONCLUSION AND FUTURE SCOPE**	82 – 83
9.	**REFERENCES**	84 – 86

1. INTRODUCTION

1. INTRODUCTION

The project Smart Health Prediction includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id.

The Smart Health prediction can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

Smart Health Prediction is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

Smart Health Prediction is designed for multi-speciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

Smart Health Prediction is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Smart Health Prediction enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes.

1.1 BACKGROUND

In the last fifteen years, HDs become still endured the leading causes of death. (WHO, 2019). In United States, Millions of human are having a HD every year so that the HD takes placed the biggest killer of people in the world. According to analyzation of WHO, twelve million people are death due to HD in worldwide. One person dies almost every 34 seconds from HD. (Patel Jaymin, 2016) Diagnosis of HDs is an essential task and yet intricate task to perform accurately and efficiently in the hospital and clinic. These things are motivated to build a Web-based HDPS application using ML algorithms. This proposed system can reserve problem by accurately predicting the presence of HD in the patient. ML is an emerging technology of AI that solved the various type of classification problem by producing accurate output. ML algorithms are applied to forecast the HD of the patient. This proposed system can be used either NB or DT algorithms by comparing their accurate result and trained time. A HDPM is built by employing one of the algorithms and deploying this model into a web application. The outcome of this project is to deliver a web-based application named HDPS that successfully and accurately predict the presence of HD of patients. This system is useful to support the decision making of doctors and healthcare members in hospitals.

With the consideration of WHO statistical facts, the most powerful causes of death globally are a HD. It seemed to the negligence of patients as well as doctors to increase a HD patient. Some of the difficulties to execute the doctor's decision and lack of application to clearly diagnosis of HD become the cause of human death. Regarding the above issues, we are proposing a web-based HDPS that is one of the best solutions to efficiently and accurately predict the HD patients. The proposed system eliminates the various testing of HD and supports the decision making of doctors. This system can accept a singleton query and display the clear output of the presence of HD level. This system is useful for any hospital and clinic to evaluate the patient getting a HD. It is reduced the number of tests and provide an efficient output of patient HD. It supports to make the decision of doctors that consulates with their patients easily. The proposed system will support the healthcare systems as well as health-related application to expand their services with efficiently and accurately providing results. It mitigates the time to checkup of doctors.

PROJECT OVERVIEW

Smart Health Prediction is a PHP MySQL which aims at minimizing the communication lag between doctors and patients. Its main concern is to notify patients about any activity in dashboard regarding their diseases so that patients can quickly access the dashboard and view the information about anything related to smart health prediction and diseases.

This project can easily Explained in terms of three entities used in the project namely:

Doctor: A doctor is someone who is qualified in medicine and treats people who are ill. Do not discontinue the treatment without consulting your doctor. A doctor is someone who has been awarded the highest academic or honorary degree by a university.

Patient: If Patient is a new user he will enter his personal details and he will user Id and password through which he can login to the system.

Admin: System admin can log into the system using his login details They manage database which contains every information regarding doctor Patient dashboard and groups They can access each and every information related to teachers and students.

1.2 PROBLEM STATEMENT

Numerous issues are faced by patients pertinent to hospitals such as being unable to provide medical services, insufficient number of qualified medical staffs, poor communication between doctors and patients, and unorganized health records and data. Eventually, these issues opportunity for hospitals to handle both their management and their duties steadily to maintain the health of every citizen and community.

Hospitals currently use a manual system for the management and maintainance of critical information the current system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information is incomplete or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

1.3 PROPOSED SYSTEM

To beat the downside of existing framework we have created smart health prediction System. We have built up a specialist framework called Smart Health Prediction framework, which is utilized for improving the task of specialists. A framework checks a patient at initial level and proposes the possible diseases. It begins with getting some information about manifestations to the patient, in the event that the framework can distinguish the fitting sickness, at that point it proposes a specialist accessible to the patient in the closest conceivable territory. On the off chance that the framework isn't sufficiently sure, it asks few questions to the patients, still on the off chance that the framework isn't sure; at that point it will show a few tests to the patient. In light of accessible total data, the framework will demonstrate the result. Here we utilize some intelligent minin methods to figure the most precise disorder that could be associated with patient's appearances and dependent on the database of a couple of patients restorative record, calculation (Naïve Bayes) is connected for mapping the side effects with conceivable diseases.This framework improves undertaking of the specialists as well as helps the patients by giving vital help at a soonest organize conceivable.

We have built up a specialist framework called Smart Health Prediction framework, which is utilized for improving the task of specialists. A framework checks a patient at initial level and proposes the possible diseases. It begins with getting some information about manifestations to the patient, in the event that the framework can distinguish the fitting sickness, at that point it proposes a specialist accessible to the patient in the closest conceivable territory. On the off chance that the framework isn't sufficiently sure, it asks few questions to the patients, still on the off chance that the framework isn't sure; at that point it will show a few tests to the patient. In light of accessible total data, the framework will demonstrate the result. This framework improves undertaking of the specialists as well as helps the patients by giving vital help at a soonest organize conceivable.

MODULES:

ADMIN LOGIN

- Admin logins to the system and manage all the functionalities of Health Care System .
- Admin can add, edit, delete and view the records of Health Centre, Appointment, Medicine, Fees.
- Admin can manage all the details of Booking, Customer, Test .
- Admin can also generate reports of Health Centre, Booking, Appointment, Customer, Medicine, Test .
- Admin can search the details of Booking, Medicine, Test .
- Admin can apply different level of filters on report of Health Centre, Customer, Medicine .
- Admin can tracks the detailed information of Booking, Appointment, Customer, Medicine

PATIENT LOGIN

- Patient Login to the system using his ID and Password.
- If Patient is a new user he will enter his personal details and he will user Id and password through which he can login to the system.
- Patient can view his personal details.
- Patient will specify the symptoms caused due to his illness. System will ask certain question regarding his illness and system predict the disease based on the symptoms specified by the patient and system will also suggest doctors based on the disease.
- Patient can search for doctor by specifying name, address or type.
- Patient will give feedback this will be reported to the admin.

DOCTOR LOGIN

- Doctor will access the system using his User ID and Password.
- Doctor can view patient's personal details.
- Doctor can add the appointment to the patient
- Doctor can view the appointment history

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

2.1 PRELIMINARY INVESTIGATION

The first and foremost strategy for development of a project starts from the thought of designing a mail enabled platform for a small firm in which it is easy and convenient of sending and receiving messages, there is a search engine, address book and also including some entertaining games. When it is approved by the organization and our project guide the first activity, i.e. preliminary investigation begins. The activity has three parts:

Request Clarification

Feasibility Study

Request Approval

REQUEST CLARIFICATION

After the approval of the request to the organization and project guide, with an investigation being considered, the project request must be examined to determine precisely what the system requires.

Here our project is basically meant for users within the company whose systems can be interconnected by the Local Area Network(LAN). In today's busy schedule man need everything should be provided in a readymade manner. So taking into consideration of the vastly use of the net in day to day life, the corresponding development of the portal came into existence.

2.2 FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are

Operational Feasibility

Economic Feasibility

Technical Feasibility

Operational Feasibility

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

Economic Feasibility

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at any time. The Virtual Private Network is to be developed using the existing resources of the organization. So the project is economically feasible.

Technical Feasibility

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform Independent environment. Java Server Pages, JavaScript, HTML, SQL server and WebLogic Server are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

FUNCTIONAL REQUIREMENTS

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs (see also software). Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases. Generally, functional requirements are expressed in the form system shall do <requirement>. The plan for implementing functional requirements is detailed in the system design. In requirements engineering, functional requirements specify particular results of a system. Functional requirements drive the application architecture of a system. A requirements analyst generates use cases after gathering and validating a set of functional requirements. The hierarchy of functional requirements is: user/stakeholder request -> feature -> use case -> business rule.

Functional requirements drive the application architecture of a system. A requirements analyst generates use cases after gathering and validating a set of functional requirements. Functional requirements may be technical details, data manipulation and other specific functionality of the project is to provide the information to the user.

NON FUNCTIONAL REQUIREMENTS

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors

The project non-functional requirements include the following.

Updating Work status.

Problem resolution.

Error occurrence in the system.

Customer requests.

Availability: A system's "availability" or "uptime" is the amount of time that is operational and available for use. It's related to is the server providing the service to the users in displaying images. As our system will be used by thousands of users at any time our system must be available always. If there are any cases of updations they must be performed in a short interval of time without interrupting the normal services made available to the users.

Efficiency: Specifies how well the software utilizes scarce resources: CPU cycles, disk space, memory, bandwidth etc. All of the above mentioned resources can be effectively used by performing most of the validations at client side and reducing the workload on server by using JSP instead of CGI which is being implemented now.

Flexibility: If the organization intends to increase or extend the functionality of the software after it is deployed, that should be planned from the beginning; it influences choices made during the design, development, testing and deployment of the system. New modules can be easily integrated to our system without disturbing the existing modules or modifying the logical database schema of the existing applications.

Portability: Portability specifies the ease with which the software can be installed on all necessary platforms, and the platforms on which it is expected to run. By using appropriate server

versions released for different platforms our project can be easily operated on any operating system, hence can be said highly portable.

Scalability: Software that is scalable has the ability to handle a wide variety of system configuration sizes. The nonfunctional requirements should specify the ways in which the system may be expected to scale up (by increasing hardware capacity, adding machines etc.). Our system can be easily expandable. Any additional requirements such as hardware or software which increase the performance of the system can be easily added. An additional server would be useful to speed up the application.

Integrity: Integrity requirements define the security attributes of the system, restricting access to features or data to certain users and protecting the privacy of data entered into the software. Certain features access must be disabled to normal users such as adding the details of files, searching etc which is the sole responsibility of the server. Access can be disabled by providing appropriate logins to the users for only access.

Usability: Ease-of-use requirements address the factors that constitute the capacity of the software to be understood, learned, and used by its intended users. Hyperlinks will be provided for each and every service the system provides through which navigation will be easier. A system that has high usability coefficient makes the work of the user easier.

Performance: The performance constraints specify the timing characteristics of the software.

3. SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

- Operating system: windows, Linux
- Processor : minimum core i3
- Ram: minimum 4 gb
- Hard disk : minimum 1TB

3.2 SOFTWARE REQUIREMENTS

The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

- Language Used : Python
- Backend Framework : Django Framework
- Web Technologies : Html, Css, Javascript
- Database : Mysql

HTML

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML is a language that is easy to write, easy to understand and highly portable. HTML is not a compiled language and is directly interpreted by a browser.HTML is the set of instructions. Each instruction is called as an element or Markup. It is used to structure and format documents for presentation on the web. HTML enhances ASCII files with markup tags that permit the display of a variety of fonts, images, and highlighting options. It also designates structural elements such as headers, lists, and paragraphs, and provides hypertext links to other documents on the Internet.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate css file, and reduce complexity and repetition in the structural content.

JAVASCRIPT

JavaScript often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web

applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

Java Script is Netscape's cross–platform, object-based scripting language for client server application. JavaScript is mainly used as a client side scripting language. This means JavaScript code is written into an HTML page. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it's up to the browser to do something with it. JavaScript can be used in other contexts than a Web browser. Netscape created server-side JavaScript as a CGI-language that can do roughly the same as Perl or ASP.

Fortunately most browsers can handle JavaScript nowadays, but of course some browsers do not support some bits of script.

Types of Java Script:

a. Navigator Java Script also called client-side Java Script.

b. Live Wire Java Script also called server-side Java Script.

Using Java Script, dynamic HTML pages can be created that process user input and maintain persistent data using special objects, files and relational databases. Browser interprets JavaScript statements embedded in an HTML page. Netscape Navigator 2.0 and Internet Explorer 3.0 versions and later recognize Java Script. Through JavaScript Live Connect functionally, application can access Java and CORBA distributed-object applications. Navigator 3.0 and later versions supports Live Connect.

Features of JavaScript (JS):

- Browser interprets JavaScript.
- JavaScript is object based and uses built-in, extensible objects and have no classes or inheritance
- JavaScript is loosely typed language
- In JavaScript object reference are checked at runtime

- JavaScript is designed to supplement the capabilities of HTML with script that are capable of responding to web pages events. JSP has access to some extent of aspects of the web browser window.
- JavaScript control browser and content but cannot draw graphics or perform networking.

Client side JavaScript features:

Client–side JavaScript has expressly been developed for use in a web browser in conjunction with HTML pages. This has certain consequences for security.

JavaScript cannot read files from or write them to the file system on the computer. This would be a clear security hazard

JavaScript cannot execute any other programs. This would also be unacceptable.

JavaScript cannot establish any connection to whatever computer, except to download a new HTML page or to send mail. This, too, would create unacceptable hazards.

The Client-Side JavaScript also has the following features:

- Controls Document's appearance and content
- Control the browser
- Interact with the HTML forms
- Interact with the user
- Read and write client state with cookies

Server- Side JavaScript Features:

- Embedded in HTML page
- Executed at the server
- Pre-complied for faster response
- Access to Server-side objects

- Encapsulation of the request

MYSQL

Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. MySQL Workbench is available on Windows, Linux and Mac OSX.

PYTHON

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

HISTORY

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL), capable of exception handling and interfacing with the Amoeba operating system. Its

implementation began in December 1989.[36] Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's *Benevolent Dictator For Life*, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker.[37] He now shares his leadership as a member of a five-person steering council. In January 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member "Steering Council" to lead the project.[41]

Python 2.0 was released on 16 October 2000 with many major new features, including a cycle-detecting garbage collector and support for Unicode.

Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible.[43] Many of its major features were backported to Python 2.6.x and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.

Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3.

Python Libraries

After Modules and Python Packages, we shift our discussion to Python Libraries. This Python Library, we will discuss Python Standard library and different libraries offered by Python Programming Language: pandas, Matplotlib, scipy, numpy, etc.

What is the Python Libraries?

We know that a module is a file with some Python code, and a package is a directory for sub packages and modules. But the line between a package and a Python library is quite blurred.

A Python library is a reusable chunk of code that you may want to include in your programs/ projects. Compared to languages like C++ or C, a Python libraries do not pertain to any specific context in Python. Here, a ‘library’ loosely describes a collection of core modules. Essentially, then, a library is a collection of modules. A package is a library that can be installed using a package manager like rubygems or npm.

3.3 INTRODUCTION TO SOFTWARE ENVIRONMENT

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

Health prediction development is the process by which new application are created for my SQL. Officially apps can be written using python, my SQL using the Android software development kit and third party tools. Environment and language support have also continued to evolve and expand since initial was released in application.

FEATURES OF OS

INTERFACE

The technical feasibility is frequently the most difficult area encountered at this stage. It is essential that the process of analysis and definition be conducted in parallel with an assessment to technical feasibility. It centers on the existing computer system (hardware, software etc.) and to what extent it can support the proposed system.

APPLICATION

The document of the Project proposal for developing a health care system This system is fully updated and we had provide some new things in it. The project aimed to build a fully functional system in order to achieve the efficiency in faster heath treatment and online consultation system.

The overall mission of system development is to make the primary treatment quickly and easily complete the Online Consultation System.

In doctor's module when a doctor login to the system, doctor can view his patient’s details and the report of that patient. Doctor can view details about the patient and what patient searched for according to their prediction. Doctor can view his personal details. Admin can add new disease details by specifying the type and symptoms of the disease into the database.

It might have happened so many times that you or someone of yours need doctor’s help immediately, but they are not available due to some reason. The Health Prediction system is an end user support and online consultation project.

The system allows user to share their symptoms and issues. Then the system processes user's symptoms to check for various illnesses that could be associated with it.

4. SYSTEM DESIGN

DESIGN INTRODUCTION

The Unified Modelling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the software system and its components. It is a graphical language which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

4.1 UML DIAGRAM

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs you draw diagram to visualize a system from different perspective, so a diagram is a projection into a system. For all but most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams, only a few diagrams , or in no diagrams at all. In theory, a diagram may contain any combination of things and relationships. In practice, however, a small number of common combinations arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system.

UML is a method for describing the system architecture in detail using the blue print. UML represents a collection of best engineering practice that has proven successful in the modeling of large and complex systems. The UML is very important parts of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the helps UML helps project teams communicate explore potential designs and validate the architectural design of the software.

UML offers a set of standardized diagram types with which complex data, processes and systems can easily be arranged in a clear, intuitive manner.

UML is neither a procedure nor a process; rather, it provides a "dictionary" of symbols

- each of which has a specific meaning. It offers diagram types for object-oriented analysis, design and programming, thereby ensuring a seamless transition from requirements placed on a system

to final implementation. Structure and system behaviour are likewise shown, thereby offering clear reference points for solution optimization.

One major aspect of UML is the ability to use diagrams as a part of project documentation. These can be utilised in various ways in the most diverse kinds of documents; for example, Use Case Diagrams used in describing functional requirements can be specified in the requirements definition. Classes or component diagrams can be used as software architecture in a design document. As a matter of principle, UML diagrams can be used in practically any technical documentation (e.g. test plans) while also serving as part of the user handbook.

1. USE CASE DIAGRAME

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

Use case diagram represents the functionality of the system. Use case focus on the behavior of the system from external point of view. Actors are external entities that interact with the system.

Use case diagram of our project

A use case diagram at its simplest is a represented of a user's interaction with the system that show the relationship between the user and the different use cases in which the is involved .A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

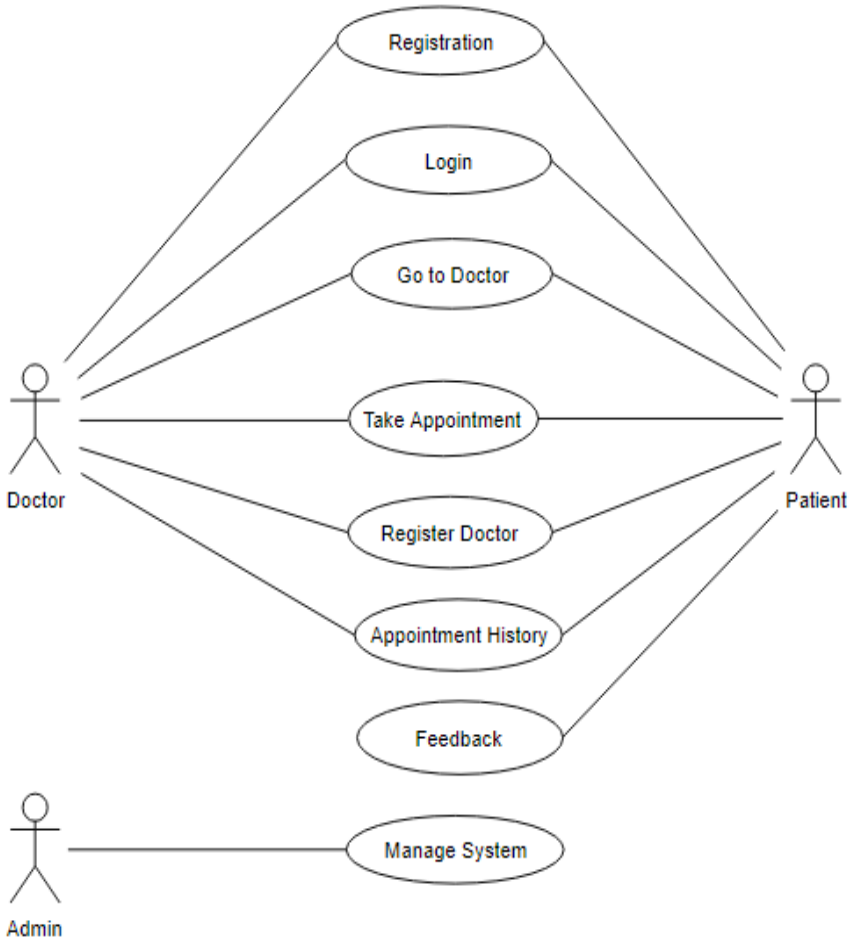


FIG 3.1 USE CASE DIAGRAM

Use cases:

A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

Actors:

An actor is a person, organization, or external system that plays a role in one or more interactions with the system.

System boundary boxes:

A rectangle is drawn around the use cases, called the system boundary box, to indicate the scope of system. Anything within the box represents functionality that is in scope and anything outside the box is not.

Four relationships among use cases are used often in practice.

Include:

In one form of interaction, a given use case may include another. "Include is a Directed Relationship between two use cases, implying that the behaviour of the included use case is inserted into the behaviour of the including use case.

The first use case often depends on the outcome of the included use case. This is useful for extracting truly common behaviours from multiple use cases into a single description. The notation is a dashed arrow from the including to the included use case, with the label "«include»". There are no parameters or return values. To specify the location in a flow of events in which the base use case includes the behaviour of another, you simply write include followed by the name of use case you want to include, as in the following flow for track order.

Extend:

In another form of interaction, a given use case (the extension) may extend another. This relationship indicates that the behaviour of the extension use case may be inserted in the extended use case under some conditions. The notation is a dashed arrow from the extension to the extended use case, with the label "«extend»". Modellers use the «extend» relationship to indicate use cases that are "optional" to the base use case.

Generalization:

In the third form of relationship among use cases, a generalization/specialization relationship exists. A given use case may have common behaviours, requirements, constraints, and assumptions with a more general use case. In this case, describe them once, and deal with it in the same way, specialized cases. The notation is a solid line ending in a hollow triangle drawn from the specialized to the more general use case (following the standard generalization notation.

Associations:

Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case.

Associations are modelled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the relationship or to indicate the primary actor within the use case.

Identified Use Cases:

The “user model view” encompasses a problem and solution from the preservative of those individuals whose problem the solution addresses. The view presents the goals and objectives of the problem owners and their requirements of the solution. This view is composed of “use case diagrams”. These diagrams describe the functionality provided by a system to external integrators. These diagrams contain actors, use cases, and their relationships.

2. DATA FLOW DIAGRAM

Level-1 DFD for Doctors

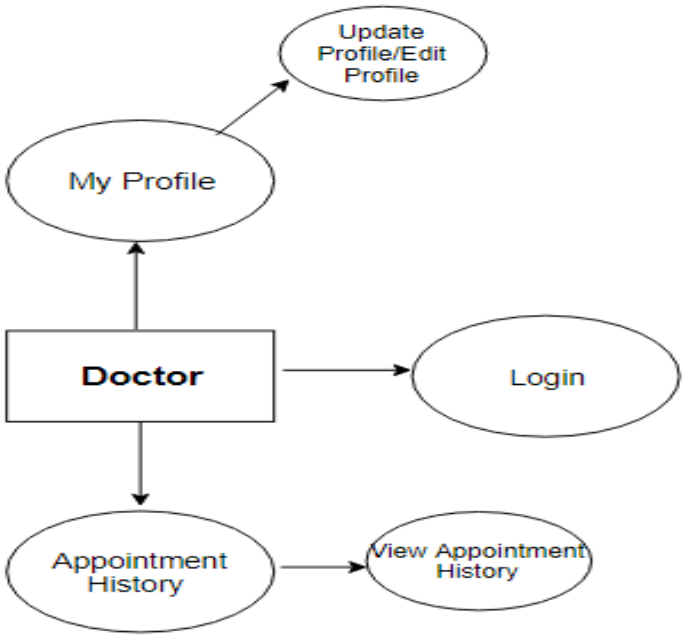


Fig3.3 Level-1 DFD

Level-1 DFD for Patients

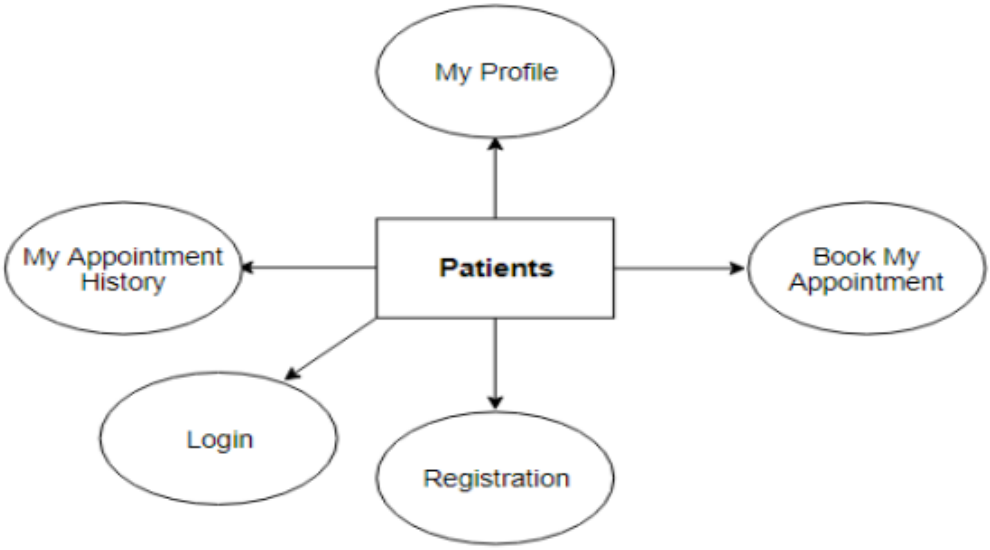


Fig 3.4 Level-1 DFD

Level-1 DFD for Admin

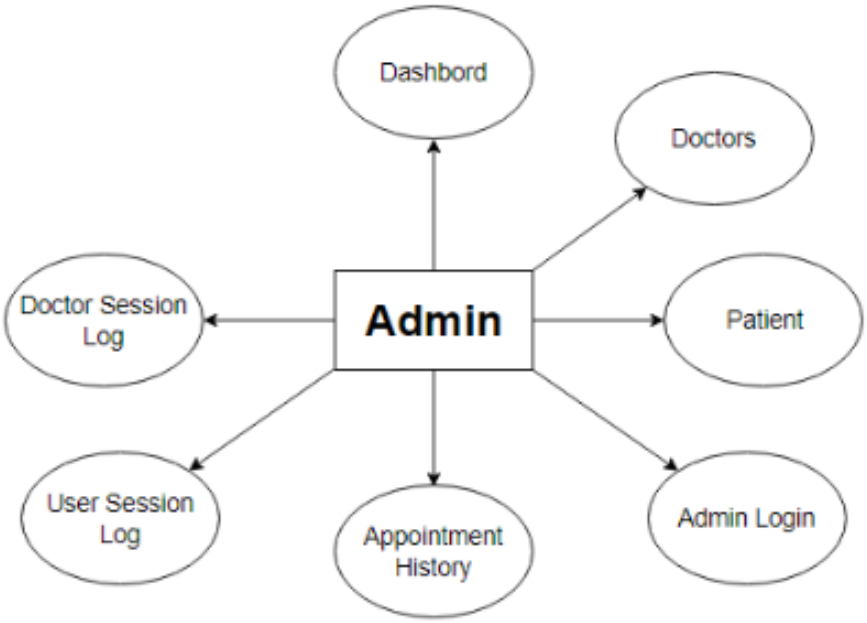


Fig 3.5 Level-1 DFD

Level-2 DFD for Doctors

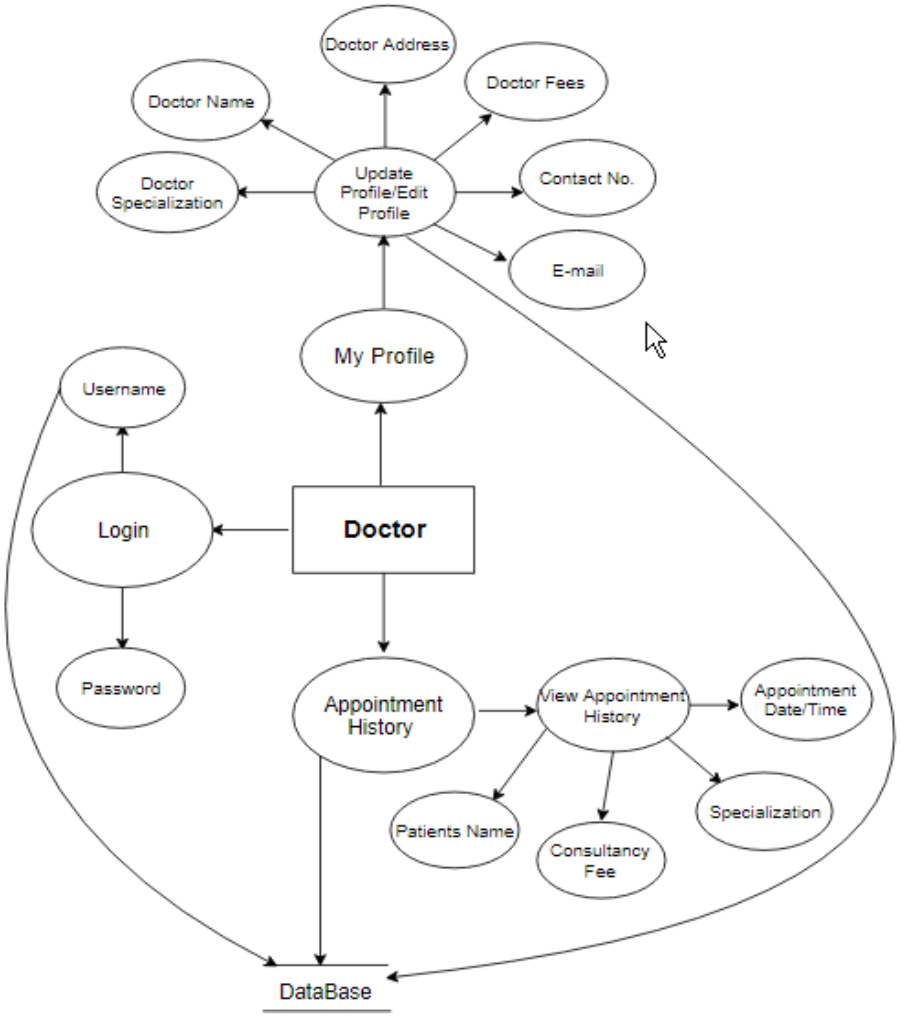


Fig 3.6 Level-2 DFD

Level-2 DFD for Patients

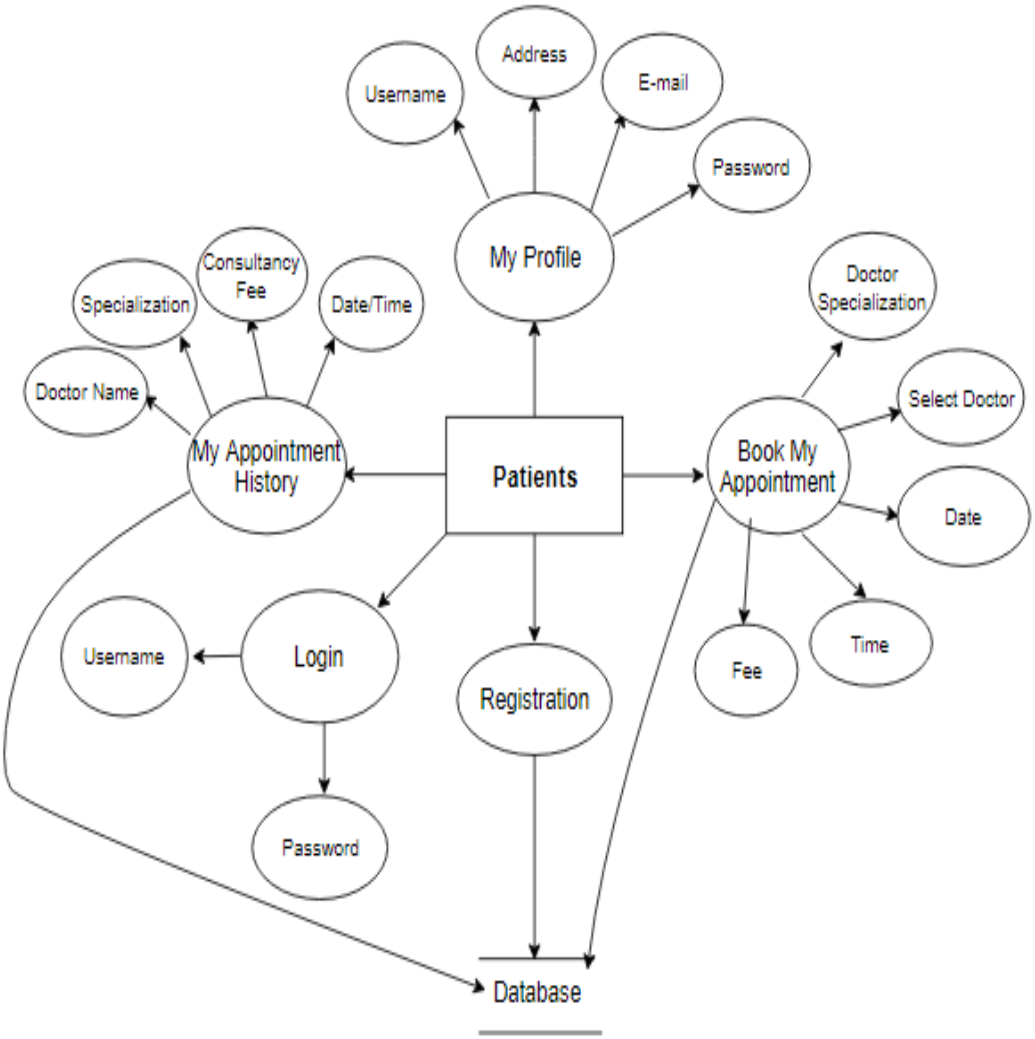


Fig 3.7 Level-2 DFD

Level-2 DFD for Admin

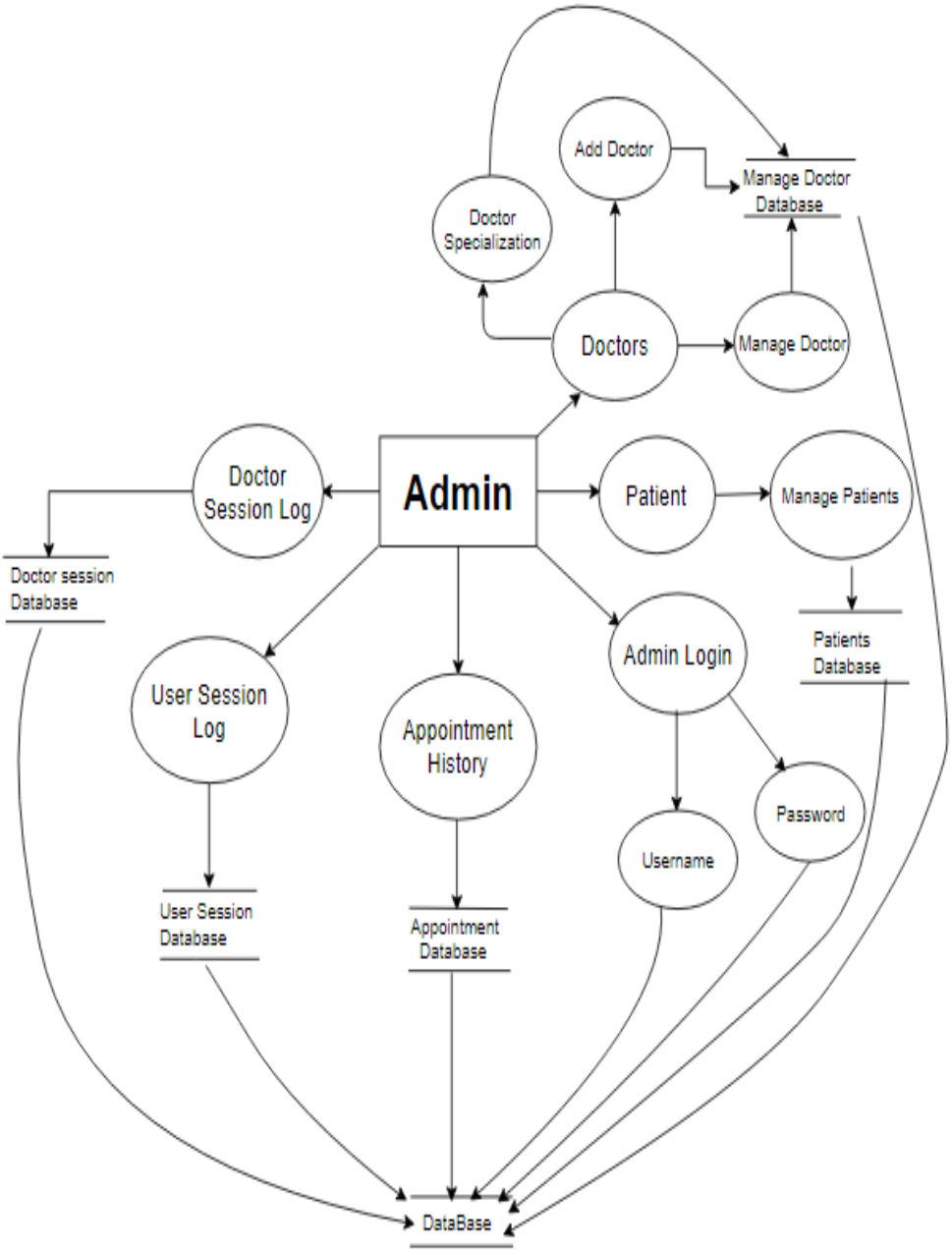
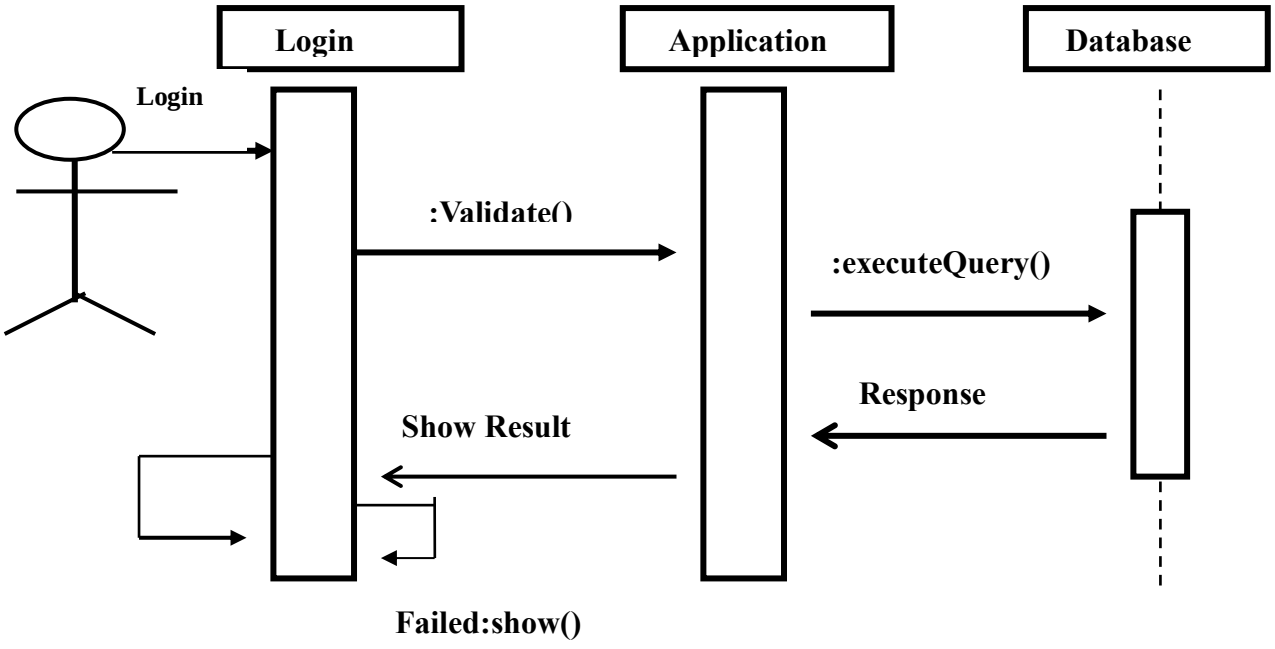


Fig 3.8 Level-2 DFD

3. SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.

Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. If the lifeline is that of an object, it demonstrates a role. Note that leaving the instance name blank can represent anonymous and unnamed instances. In order to display interaction, messages are used. These are horizontal arrows with the message name written above them. Solid arrows with full heads are synchronous calls, solid arrows with stick heads are asynchronous calls and dashed arrows with stick heads are return messages.



5. IMPLEMENTATION

5.1 IMPLEMENTATION

IMPLEMENTATION

An important issue for the development of a project is the selection of suitable front-end and back-end. When we decided to develop the project we went through an extensive study to determine the most suitable platform that suits the needs of the organization as well as helps in development of the project.

A study of resource availability that may affect the ability to achieve an acceptable system This evaluation determines whether the technology needed for the proposed system is available or not.

A more sophisticate register maintenance for various Patient Information, Doctor diary, Immunization Details and a good system for writing bill amount employees and stock availed for the customers can be maintained at central place.

Adequate staff may be maintained so that updations are made at the very moment at the same time. Proper person for proper work should be made responsible so that a better efficiency could be achieved. This needs a lot of work force.

Another alternative solution can be used of computer based batch system for maintaining the information regarding purchase details, customers and employees. A batch system refers to a system in which data is processed in a periodical basis.

The batch system is able to achieve most of the goals and sub goals. But a batch system data is processed in sequential basis. Therefore batch system is not suggested.

Another alternative solution can be used of computer based batch system for maintaining the information regarding purchase details, customers and employees. A batch system refers to a system in which data is processed in a periodical basis.

The batch system is able to achieve most of the goals and sub goals. But a batch system data is processed in sequential basis. Therefore batch system is not suggested.

- Careful planning
- Investigation of system and constraints

- Design of methods to achieve the changeover
- Training of the staff in the changeover phase

IMPLEMENTATION PROCEDURES

The designing and implementation of Online Health Prediction System with the gathering of requirements and examine the background of the hospital management.

Although the current system is a manual and file based one, we realize that the system we are going to build must give the solutions for wastage of time and space which affects the efficiency of the daily activities performed at the hospital.

USER TRAINING

One of the alternative solutions is the improvement of the manual system. Anything, which can be done by using automated methods, can be done manually. But the question arises how to perform thing manually in a sound manner. Following are some suggestions, which can be useful in the manual system.

We have decided to use prototyping model to develop Health Prediction System. Therefore the system is developed in increments so that it can readily be modified in response to end-user and customer feedback. A prototype is built with basic and critical attributes.

TRAINING ON THE APPLICATION SOFTWARE

All this work is done manually by the receptionist and other operational staff and lot of papers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can't remember them at that time.

We have decided to use prototyping model to develop Health Prediction System. Therefore the system is developed in increments so that it can readily be modified in response to end-user and customer feedback. A prototype is built with basic and critical attributes.

The proposed system will give the minute information, as a result the performance is improved which in turn may be expected to provide increased profits.

OPERATIONAL DOCUMENT

This feasibility checks whether the system can be developed with the available funds. The Smart health prediction does not require enormous amount of money to be developed. This can be done economically if planned judicially, so it is economically feasible. The cost of project depends upon the number of required.

Although the current system is a manual and file based one, we realize that the system we are going to build must give the solutions for wastage of time and space which affects the efficiency of the daily activities performed at the hospital.

SYSTEM MAINTENANCE

An important issue for the development of a project is the selection of suitable front-end and back-end. When we decided to develop the project we went through an extensive study to determine the most suitable platform that suits the needs of the organization as well as helps in development of the project.

The designing and implementation of Online Health Prediction System with the gathering of requirements and examine the background of the hospital management.

Although the current system is a manual and file based one, we realize that the system we are going to build must give the solutions for wastage of time and space which affects the efficiency of the daily activities performed at the hospital.

During feasibility analysis for this project, following primary areas of interest are to be considered. Investigation and generating ideas about a new system does this.

This is the document of the Project proposal for developing a health care system. This system is fully updated and we had provide some new things in it. The project aimed to build a fully functional system in order to achieve the efficiency in faster heath treatment and online consultation system. The overall mission of system development is to make the primary treatment quickly and easily complete the Online Consultation System.

A study of resource availability that may affect the ability to achieve an acceptable system This evaluation determines whether the technology needed for the proposed system is available or not.

The system allows user to share their symptoms and issues. Then the system processes user's symptoms to check for various illnesses that could be associated with it.

5.2 CODE

The purpose of code is to facilitate the identification. Retrieval of the items and information A code is an oriented collection of symbols design to provides .Unique identification of the entry or attribute. Code is built with manually exclusive feature. Codes in all cases specify object which are physical or on performing characteristics. They are used to give optimal distraction and other information. Codes insure that only one value of the code with single meaning is correctly applied to give entity or attributes as described in various wave. Code can also be design in manner easily understood and applied by the user.

HOME PAGE CODING

```
<!DOCTYPE html>
{% load static %}
<html lang="zxx">
<head>
    <title>Health Prediction System</title>
    <!-- Meta tag Keywords -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta charset="UTF-8" />
    <meta name="keywords" />
    <script>
        addEventListener("load", function () {
            setTimeout(hideURLbar, 0);
```

```
		}, false);


		function hideURLbar() {
			window.scrollTo(0, 1);
		}
	</script>
	<link rel="stylesheet" href="{% static 'css/bootstrap.css' %}">
	<!-- Bootstrap-Core-CSS -->
	<link href="{% static 'css/css_slider.css' %}" type="text/css" rel="stylesheet" media="all">
	<!-- banner slider -->
	<link rel="stylesheet" href="{% static 'css/style.css' %}" type="text/css" media="all" />
	<!-- Style-CSS -->
	<link href="{% static 'css/font-awesome.min.css' %}" rel="stylesheet">
	<!-- Font-Awesome-Icons-CSS -->
	<!-- //Custom-Files -->
	<!-- Web-Fonts -->
	<link href="//fonts.googleapis.com/css?family=Quicksand:300,400,500,700" rel="stylesheet">
	<link href="//fonts.googleapis.com/css?family=Mukta:200,300,400,500,600,700,800&amp;subset=devanagari,latin-ext" rel="stylesheet">
<style>
<!-- //Web-Fonts -->
```

```
.li2{
        border:1px solid red;
        padding:8px;
        }
</style>

<link rel='stylesheet' type='text/css'
href="https://cdn.datatables.net/1.10.21/css/jquery.dataTables.min.css">
<link rel='stylesheet' type='text/css'
href="https://cdn.datatables.net/buttons/1.6.2/css/buttons.dataTables.min.css">

<script src="https://code.jquery.com/jquery-3.5.1.js"></script>
<script src="https://cdn.datatables.net/1.10.21/js/jquery.dataTables.min.js"></script>
<script src="https://cdn.datatables.net/buttons/1.6.2/js/dataTables.buttons.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/pdfmake.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/vfs_fonts.js"></script>
<script src="https://cdn.datatables.net/buttons/1.6.2/js/buttons.html5.min.js"></script>

<script>
$(document).ready(function() {
    $('#example').DataTable( {
        dom: 'Bfrtip',
```

```
        buttons: [
            'copyHtml5',
            'excelHtml5',
            'csvHtml5',
            'pdfHtml5'
        ]
    } );
} );
</script>
</head>

<body>
<!-- main -->
<div id="home" style="margin-bottom:2%">
<!-- top header -->
<header>
<div class="top-bar py-3">
<div class="container">
<div class="row">
<div class="col-xl-6 col-lg-6 col-md-8 top-social-agile text-lg-left text-center">
<div class="row">
<div class="col-5 header-top_w3layouts">
<p class="text-bl">
```

```
<span class="fa fa-map-marker mr-2"></span>Indrapuri Sec-C, Bhopal

</p>

</div>

<div class="col-3 header-top_w3layouts">

        <p class="text-bl">

        <span class="fa fa-phone mr-2"></span>+1 000263676

        </p>

</div>

<!-- social icons -->

<ul class="col-4 top-right-info">

<li>

        <a href="#">

        <span class="fa fa-facebook-f"></span>

        </a>

</li>

<li class="mx-3">

        <a href="#">

        <span class="fa fa-twitter"></span>

        </a>

</li>

<li>

        <a href="#">

        <span class="fa fa-google-plus"></span>
```

```
        </a>
</li>
<li class="ml-3">
        <a href="#">
        <span class="fa fa-pinterest-p"></span>
        </a>
</li>
</ul>
<!-- //social icons -->
</div>
</div>
<div class="col-xl-7 col-lg-6 col-md-4 top-social-agile text-md-right text-center mt-md-0 mt-2">
<div class="row">
<div class="offset-xl-6 offset-lg-4">
</div>
<div class="col-xl-3 col-lg-4 col-6 top-w3layouts p-md-0 text-right">
        <!-- login -->
        <a href="login.html" class="login-button-2 text-uppercase text-bl">
        </a>
        <!-- //login -->
</div>
<div class="col-xl-3 col-lg-4 col-6 header-w3layouts text-md-right text-left">
        <!-- register -->
```

```
        <a href="register.html" class="login-button-2 text-uppercase text-bl">
        </a>
        <!-- //register -->
</div>
</div>
</div>
</div>
</div>
</div>
</header>
<!-- //top header -->
<!-- second header -->
<div class="main-top" style="background:Green">
<div class="container-fluid">
<div class="header d-md-flex justify-content-between align-items-center py-3">
<!-- logo -->
<div id="logo" style="margin-left:13%">
        <h1>
        <a href="index.html">
        <span class="fa fa-user-md mr-2"></span>
        <span class="logo-sp">Health</span> Prediction
        </a>
        </h1>
```

```
</div>

<!-- //logo -->

<!-- nav -->

{% if request.user.is_staff %}

<div class="nav_w3ls" style="margin-right:4%">

<nav>

<label for="drop1" class="toggle">Menu</label>

<input type="checkbox" id="drop1" />

<ul class="menu" style="padding:15px;width:100%">

<li class="mx-lg-4 mx-md-3 my-md-0 my-2"><a href="{% url 'admin_home' %}"
class="active">Home</a></li>

<li>

<label for="drop-2" class="toggle toogle-2"> <span class="fa fa-angle-down" aria-
hidden="true"></span>

</label>

<a href="#"> Doctor <span class="fa fa-angle-down" aria-hidden="true"></span></a>

<ul style="margin-top:2%;width:40%;margin-right:0%">

<li style="width:100%;"><a href="{% url 'add_doctor' %}" class="drop-text">Add
Doctor</a></li>

<li style="width:100%;"><a href="{% url 'view_doctor' %}" class="drop-text">View
Doctor</a></li>

</ul>

</li>

<li class="mx-lg-4 mx-md-3 my-md-0 my-2">
```

```
<label for="drop-2" class="toggle toogle-2"> <span class="fa fa-angle-down" aria-hidden="true"></span>

</label>

<a href="#"> Disease <span class="fa fa-angle-down" aria-hidden="true"></span></a>

<ul style="margin-top:2%;width:40%;margin-right:4%">

<li style="width:100%;"><a href="{% url 'add_disease' %}" class="drop-text">Add Disease</a></li>

<li style="width:100%;"><a href="{% url 'view_disease' %}" class="drop-text">View Disease</a></li>

</ul>

</li>

<li><a href="{% url 'view_patient' %}">View Patient</a></li>

<li class="mx-lg-4 mx-md-3 my-md-0 my-2"><a href="{% url 'view_feedback' %}">View Feedback</a></li>

<li><a href="{% url 'view_notify' %}"><i class="fa fa-bell"></i> </a></li>

</ul>

</nav>

</div>

<div class="nav_w3ls" style="margin-right:3%">

<nav>

<label for="drop2" class="toggle">Menu</label>

<input type="checkbox" id="drop2" />

<ul class="menu">

<li class="mx-lg-4 mx-md-3 my-md-0 my-2" style="background:red;padding:8px">
```

```
<!-- First Tier Drop Down -->

<label for="drop-2" class="toggle toogle-2"> <span class="fa fa-angle-down" aria-hidden="true"></span>

</label>

<a href="#"><i class="fa fa-user"></i> {{request.user.username}} <span class="fa fa-angle-down" aria-hidden="true"></span></a>

<ul style="margin-top:10%;width:80%;margin-right:5%">

<li style="width:100%;">

<a href="{% url 'change_password' %}" class="drop-text">Password</a>

</li>

<li style="width:100%;">

<a href="{% url 'logout' %}" class="drop-text">Logout</a>

</li>

</ul>

</li>

</ul>

</nav>

</div>

{% else %}

{% if request.user.is_authenticated %}

{% ifequal error "pat" %}

<div class="nav_w3ls" style="margin-left:5%;font-weight:bold">

<nav>
```

```
<label for="drop3" class="toggle">Menu</label>

<input type="checkbox" id="drop3" />

<ul class="menu" style="padding:15px;width:100%">

<li><a href="{% url 'patient_home' %}" class="active">Home</a></li>

<li class="mx-lg-4 mx-md-3 my-md-0 my-2"><a href="{% url 'search_doctor' %}">Search Doctor</a></li>

<li><a href="{% url 'predict_disease' '0' %}">Search Disease</a></li>

<li class="mx-lg-4 mx-md-3 my-md-0 my-2"><a href="{% url 'profile_doctor' %}">My Detail</a></li>

<li><a href="{% url 'sent_feedback' %}">Feedback</a></li>

</ul>

</nav>

</div>

<div class="nav_w3ls" style="margin-right:3%">

<nav>

<label for="drop4" class="toggle">Menu</label>

<input type="checkbox" id="drop4" />

<ul class="menu">

<li class="mx-lg-4 mx-md-3 my-md-0 my-2" style="background:red;padding:8px">

<!-- First Tier Drop Down -->

<label for="drop-3" class="toggle toogle-3"> <span class="fa fa-angle-down" aria-hidden="true"></span>

</label>
```

```
<a href="#"><i class="fa fa-sign-in"></i> Hello,{{request.user.username}} <span class="fa fa-angle-down" aria-hidden="true"></span></a>

<ul style="margin-top:8%;width:80%;margin-right:5%">

<li style="width:100%;"><a href="{% url 'change_password' %}" class="drop-text"> Password</a></li>

<li style="width:100%;"><a href="{% url 'logout' %}" class="drop-text"> Logout</a></li>

</ul>

</li>

</ul>

</nav>

</div>

{% else %}

<div class="nav_w3ls" style="margin-right:4%">

<nav>

<label for="drop5" class="toggle">Menu</label>

<input type="checkbox" id="drop5" />

<ul class="menu" style="padding:15px;width:100%">

<li><a href="{% url 'doctor_home' %}" class="active">Home</a></li>

<li class="mx-lg-4 mx-md-3 my-md-0 my-2">

<a href="{% url 'profile_doctor' %}">My Detail</a></li>

<li><a href="{% url 'notification' %}">Notification</a></li>

</ul>

</nav>
```

```
</div>

<div class="nav_w3ls" style="margin-right:3%">

<nav>

<label for="drop6" class="toggle">Menu</label>

<input type="checkbox" id="drop6" />

<ul class="menu">

<li class="mx-lg-4 mx-md-3 my-md-0 my-2" style="background:red;padding:8px">

<!-- First Tier Drop Down -->

<label for="drop6" class="toggle toogle-2"> <span class="fa fa-angle-down" aria-hidden="true"></span>

</label>

<a href="#"><i class="fa fa-user"></i> {{request.user.username}} <span class="fa fa-angle-down" aria-hidden="true"></span></a>

<ul style="margin-top:10%;width:80%;margin-right:5%">

<li style="width:100%;"><a href="{% url 'change_password' %}" class="drop-text">Password</a></li>

<li style="width:100%;"><a href="{% url 'logout' %}" class="drop-text">Logout</a></li>

</ul>

</li>

</ul>

</nav>

</div>

{% endifequal %}
```

```
{% else %}
<div class="nav_w3ls" style="margin-right:4%">
<nav>
<label for="drop7" class="toggle">Menu</label>
<input type="checkbox" id="drop7" />
<ul class="menu" style="padding:15px;width:100%">
<li><a href="{% url 'home' %}" class="active">Home</a></li>
<li class="mx-lg-4 mx-md-3 my-md-0 my-2">
<a href="{% url 'about' %}">About Us</a></li>
<li><a href="{% url 'gallery' %}">Gallery</a></li>
<li style="padding-left:20px"><a href="{% url 'contact' %}">Contact Us</a></li>
</ul>
</nav>
</div>
<div class="nav_w3ls" style="margin-right:3%">
<nav>
<label for="drop" class="toggle">Menu</label>
<input type="checkbox" id="drop" />
<ul class="menu">
<li class="mx-lg-4 mx-md-3 my-md-0 my-2" style="background:red;padding:8px">
<!-- First Tier Drop Down -->
<label for="drop-2" class="toggle toogle-2"> <span class="fa fa-angle-down" aria-
hidden="true"></span>
```

```
</label>
<a href="#"><i class="fa fa-sign-in"></i> Login <span class="fa fa-angle-down" aria-hidden="true"></span></a>
<ul style="margin-top:10%;width:80%;margin-right:5%">
<li style="width:100%;"><a href="{% url 'login_admin' %}" class="drop-text">Admin</a></li>
<li style="width:100%;"><a href="{% url 'login' %}" class="drop-text">User</a></li>
</ul>
</li>
</ul>
</nav>
</div>
{% endif %}
{% endif %}
<!-- //nav -->
</div>
</div>
</div>
<!-- //second header -->
<!-- banner -->
{% block body %}
{% endblock %}
<!-- //banner -->
</div>
```

```
<!-- //main -->

<!-- footer bottom -->

<!-- copyright -->

<div class="copy-w3pvt" style="background:green;color:white">

<div class="container py-3">

<div class="row">

<div class="col-lg-7 w3ls_footer_grid1_left text-lg-left text-center">

<p style="color:white">&copy; 2020 Health Prediction. All rights reserved | Design by

Pankaj Panjwani</a>

</p>

</div>

<div class="col-lg-5 w3ls_footer_grid_left1_pos text-lg-right text-center mt-lg-0 mt-2">

<ul style="color:white">

<li>

<a href="#" style="color:white" class="facebook">

<span class="fa fa-facebook-f mr-2"></span>Facebook</a>

</li>

<li class="mx-3">

<a href="#" style="color:white" class="twitter">

<span class="fa fa-twitter mr-2"></span>Twitter</a>

</li>

<li>

<a href="#" style="color:white" class="google">
```

```
<span class="fa fa-google-plus mr-2"></span>Google Plus</a>

</li>

</ul>

</div>

</div>

</div>

</div>

<!-- //copyright -->

<!-- //footer bottom -->

<!-- move top icon -->

<a href="#home" style="color:white" class="move-top text-center"></a>

<!-- //move top icon -->

</body>

</html>
```

USER REGISTRATION PAGE CODING

```
{% extends 'index.html' %}

{% load static %}

{% block body %}

<!-- register -->

{% ifequal error "create" %}
```

```
<script>
    alert('Registration Successfull');
    window.location="{% url 'login' %}";
</script>
{% endifequal %}
<section class="logins py-5">
<div class="container py-xl-5 py-lg-3">
<div class="title-section mb-md-5 mb-4">
<h6 class="w3ls-title-sub"></h6>
<h3 class="w3ls-title text-uppercase text-dark font-weight-bold">Register Now</h3>
</div><hr/>
<div class="login px-sm-12" style="width:100%">
<form action="" method="post" enctype="multipart/form-data">
{% csrf_token %}
<div class="form-group row">
<div class="col-md-6">
<label>First Name</label>
<input type="text" class="form-control" name="fname" placeholder="First Name" required="">
</div>
<div class="col-md-6">
<label>Last Name</label>
<input type="text" class="form-control" name="lname" placeholder="Last Name" required="">
</div>
```

```
</div>
<div class="form-group row">
<div class="col-md-6">
<label>Username</label>
<input type="text" class="form-control" name="uname" placeholder="Username" required="">
</div>
<div class="col-md-6">
<label>Password</label>
<input type="password" class="form-control" name="pwd" placeholder="Password"
required="">
</div>
</div>
<div class="form-group row">
<div class="col-md-6">
<label>Email</label>
<input type="email" class="form-control" name="email" placeholder="Enter Email"
required="">
</div>
<div class="col-md-6">
<label>Contact</label>
<input type="text" class="form-control" name="contact" placeholder="Enter Contact"
required="">
</div>
```

```
</div>
<div class="form-group row">
<div class="col-md-6">
<label>Date Of Birth</label>
<input type="date" class="form-control" name="dob" placeholder="" required="">
</div>
<div class="col-md-6">
<label>Image</label>
<input type="file" class="form-control" name="image" required="">
</div>
</div>
<div class="form-group row">
<div class="col-md-6">
<label class="mb-2">Address</label>
<input type="text" class="form-control" name="add" id="password1" placeholder="Enter Address" required="">
</div>
<div class="col-md-6">
<label>User Type</label>
<div class="form-control">
Patient <input type="radio" placeholder="Patient" name="type" style="margin-right:4%" required="" value="Patient">
Doctor <input type="radio" placeholder="Patient" name="type" required="" value="Doctor">
```

```
</div>

</div>

</div>

<button type="submit" class="btn submit mt-4">Register</button>

<p class="text-center mt-5">

<a href="#">By clicking Register, I agree to your terms</a>

</p>

</form></div></div></section><!-- //register -->

{% endblock %}
```

USER LOGIN PAGE CODING

```
{% extends 'index.html' %}

{% load static %}

{% block body %}

{% ifequal error "pat1" %}

<script>

    alert('logged in successfully');

    window.location="{% url 'patient_home' %}";

</script>

{% endifequal %}

{% ifequal error "notmember" %}
```

```
<script>
    alert('Your information verification is pending.plz login after sometimes');
    window.location=('');
</script>
{% endifequal %}
{% ifequal error "pat2" %}
<script>
    alert('logged in successfully');
    window.location="{% url 'doctor_home' %}";
</script>
{% endifequal %}
{% ifequal error "not" %}
<script>
    alert('Username & Password are not Matching');
</script>
{% endifequal %}
<!-- login -->
<section class="logins py-5">
        <div class="container py-xl-5 py-lg-3">
                <div class="title-section mb-md-5 mb-4">
                        <h6 class="w3ls-title-sub"></h6>
                        <h3 class="w3ls-title text-uppercase text-dark font-weight-bold">Login
Now</h3>
```

```
                </div><hr/>
<div class="login px-sm-4 mx-auto mw-100 login-wrapper">
<form class="login-wrapper" action="" method="post">
{% csrf_token %}
<div class="form-group">
<label>Username</label>
<input type="text" class="form-control" name="uname" placeholder="Enter Username" required="">
<small id="emailHelp" class="form-text text-muted">We'll never share your Detail with anyone else.</small>
</div>
<div class="form-group">
<label>Password</label>
<input type="password" class="form-control" name="pwd" placeholder="Enter Your Password" required="">
</div>
<button type="submit" class="btn submit mt-4">Login</button>
<p class="text-center mt-5">
<a href="{% url 'signup' %}"> Don't have an Account?  Register here</a>
</p>
</form>
</div>
</div>
```

```
</section>

<!-- //login -->

{% endblock %}
```

DISEASE PREDICTION PAGE CODING

```
{% extends 'index.html' %}

{% load static %}

{% block body %}

<section class="logins py-5">

<div class="container py-xl-5 py-lg-3">

<div class="title-section mb-md-5 mb-4">

<h6 class="w3ls-title-sub" style="color:red">

Note :- Please not refreshing this page otherwise you will not get Actual Prediction.</h6>

<h3 class="w3ls-title text-uppercase text-dark font-weight-bold">Disease Prediction</h3>

</div><hr/>

<div class="login px-sm-12" style="width:100%">

<form action="{% url 'predict_disease' '0' %}" method="post" enctype="multipart/form-data">

{% csrf_token %}

<div class="form-group">

Please enter a symptom(anyone symptom,leave no blank spaces after and before it.)

<input type="text" name="sym" required="">
```

```
 </div>
<center> <button type="submit" class="btn submit mt-4">Search</button>
</center></form>
</div>
{% ifequal terror "start" %}
<h5> Are you feeling any of these symptoms too?</h5>
<div class="login px-sm-12" style="width:100%">
{% for i in li %}
<h4 style="color:red">{{i}}</h4>
{% endfor %}
<form action="{% url 'predict_disease' '0' %}" method="post" enctype="multipart/form-data">
{% csrf_token %}
<div class="form-group row">
<div class="col-md-12">
<label>Please Select</label>
<select class="form-control" name="sym" required>
<option value="">------Select any of these -----</option>
{% for i in li %}
 <option value="{{i}}">{{i}}</option>
{% endfor %}
</select>
</div>
</div>
```

```
<div class="row">
                <div class="col-md-2">
                    <button type="submit" class="btn submit mt-4">Next</button>
                </div>
                <div class="col-md-10">
                    <button class="btn submit mt-4"><a href="{% url 'predict_disease' 'None' %}" style="color:white">I have none of the Above Symptom</a></button>
                </div>
            </div>
        </form>
</div>
{% else %}
{% ifequal terror 'End' %}
<h4 style="color:blue;margin:2%" align="center">"Analysis Complete"</h4>
<div class="form-group">
                <h3 align="center">You have suspected Disease  :
                "<span style="color:red;font-weight:bold">{{dis.name}}</span>" </h3>
</div>
 <div class="container-fluid" style="width:90%;margin-top:3%">
            <div class="container-fluid">
            <h1 align="center" style="font-weight:bold;font-family : 'Monotype Corsiva' ; color : #E6120E ;margin-top:4%">You may contact this Doctor</h1>
            </div><hr>
```

```
<table id="example" class="display" style="width:100%">
<thead>
<tr>
        <th>#</th>
        <th>Image</th>
        <th>Full Name</th>
        <th>Email</th>
        <th>Contact</th>
        <th>Address</th>
</tr>
</thead>
<tbody>
    {% for i in doc %}
        <tr>
                <td>{{forloop.counter}}</td>
                <td><img src="{{i.image.url}}" style="width:80px;height:90px"></td>
                <td>{{i.user.first_name}} {{i.user.last_name}}</td>
                <td>{{i.user.email}}</td>
                <td>{{i.contact}}</td>
                <td>{{i.address}}</td>
        </tr>
    {% endfor %}
</tbody>
```

```
</table>

</div>

{% endifequal %}

{% endifequal %}

</div>

</section>

{% endblock %}
```

SEARCH DOCTOR PAGE CODING

```
{% extends 'index.html' %}

{% load static %}

{% block body %}

<div class="container" style="margin-top:8%">

        <h6 class="w3ls-title-sub"></h6>

        <h3 class="w3ls-title text-uppercase text-dark font-weight-bold">Search Doctor</h3>

        <hr/>

        <div class="login px-sm-12" style="width:100%">

        <form action="" method="post" enctype="multipart/form-data">

        {% csrf_token %}

        <div class="form-group row">

        <div class="col-md-12">
```

```
        <label>Select By</label>
        <select class="form-control" name="type" style="border:1px solid lightgray"
required="">
                <option value="Name">Name</option>
                <option value="Type">Type</option>
                <option value="Address">Address</option>
        </select>
        </div>
</div>
<div class="form-group row">
<div class="col-md-12">
        <label>Write Text</label>
        <input type="text" class="form-control" name="tex" placeholder="Enter Contact"
required="">
</div>
</div>
<button type="submit" class="btn submit mt-4">Search</button>
</form>
</div>
</div>
{% ifequal t "Name" %}
<div class="container-fluid" style="width:90%;margin-top:0%">
 <div class="container-fluid">
```

```
<h1 align="center" style="font-weight:bold;font-family : 'Monotype Corsiva' ; color : #E6120E
;margin-top:4%">View Doctor Detail</h1>
</div><hr>
<table id="example" class="display" style="width:100%">
<thead>
        <tr>
                <th>#</th>
                <th>Full Name</th>
                <th>Email</th>
                <th>Contact</th>
                 <th>Address</th>
                 <th>Category</th>
        </tr>
</thead>
<tbody>
{% for i in doc %}
{% if i.user.id in li %}
        <tr>
                        <td>{{forloop.counter}}</td>
                        <td>{{i.user.first_name}} {{i.user.last_name}}</td>
                        <td>{{i.user.email}}</td>
                        <td>{{i.contact}}</td>
                        <td>{{i.address}}</td>
```

```
                        <td>{{i.category}}</td>
            </tr>
{% endif %}
{% endfor %}
</tbody>
</table>
</div>
{% else %}
        {% ifequal t "Type" %}
        <div class="container-fluid" style="width:90%;margin-top:%">
            <div class="container-fluid">
            <h1 align="center" style="font-weight:bold;font-family : 'Monotype Corsiva' ; color
: #E6120E ;margin-top:4%">View Doctor Detail</h1>
             </div><hr>
            <table id="example" class="display" style="width:100%">
            <thead>
                        <tr>
                          <th>#</th>
                          <th>Full Name</th>
                          <th>Email</th>
                          <th>Contact</th>
                          <th>Address</th>
                          <th>Category</th>
```

```
                                                                    </tr>
                    </thead>
  <tbody>
        {% for i in doc %}
                <tr>
                    <td>{{forloop.counter}}</td>
                    <td>{{i.user.first_name}} {{i.user.last_name}}</td>
                    <td>{{i.user.email}}</td>
                    <td>{{i.contact}}</td>
                    <td>{{i.address}}</td>
                    <td>{{i.category}}</td>
                </tr>
      {% endfor %}
</tbody>
</table>
</div></div>
{% else %}
        {% ifequal t "Address" %}
                    <div class="container-fluid" style="width:90%;margin-top:%">
              <div class="container-fluid">
                 <h1 align="center" style="font-weight:bold;font-family : 'Monotype Corsiva' ;
color : #E6120E ;margin-top:4%">View Doctor Detail</h1>
                  </div><hr>
```

```
<table id="example" class="display" style="width:100%">
        <thead>
            <tr>
                <th>#</th>
                <th>Full Name</th>
                <th>Email</th>
                <th>Contact</th>
                <th>Address</th>
                <th>Category</th>
            </tr>
      </thead>
<tbody>
    {% for i in doc %}
        <tr>
            <td>{{forloop.counter}}</td>
            <td>{{i.user.first_name}} {{i.user.last_name}}</td>
            <td>{{i.user.email}}</td>
            <td>{{i.contact}}</td>
            <td>{{i.address}}</td>
            <td>{{i.category}}</td>
        </tr>
    {% endfor %}
</tbody>
```

```
</table>

</div>

{% endifequal %}

{% endifequal %}

{% endifequal %}

{% endblock %}
```

ADD DISEASE DETAILS PAGE

```
{% extends 'index.html' %}

{% load static %}

{% block body %}

        <!-- register -->

{% ifequal error "create" %}

<script>

   alert('Add a new Disease Successfully');

   window.location="{% url 'view_disease' %}";

</script>

{% endifequal %}

<section class="logins py-5">

                <div class="container py-xl-5 py-lg-3">

                        <div class="title-section mb-md-5 mb-4">
```

```
                                <h6 class="w3ls-title-sub"></h6>

                                <h3 class="w3ls-title text-uppercase text-dark font-weight-
bold">Add Disease</h3>

                            </div><hr/>

<div class="login px-sm-12" style="width:100%">

<form action="" method="post" enctype="multipart/form-data">

{% csrf_token %}

<div class="form-group row">

        <div class="col-md-12">

        <label>Disease Name</label>

        <input type="text" class="form-control" name="d_name" placeholder="Disease Name"
required="">

        </div>

</div>

<div class="form-group row">

        <div class="col-md-12">

        <label>Write Symptom Of Disease (Seperated by ",")</label>

        <textarea class="form-control" name="sym" placeholder="Symptom Of Disease"
required="">

        </textarea>

        </div>

</div>

<div class="form-group row">
```

```
<div class="col-md-12">

<label>Select Type of Disease</label>

<select class="form-control" name="type" style="border:1px solid lightgray" required="">

{% for i in type  %}

<option value="{{i.name}}">{{i.name}}</option>

{% endfor %}

</select>

</div>

</div>

<button type="submit" class="btn submit mt-4">Register Disease</button>

</form>

</div>

</div>

</section>

<!-- //register -->

{% endblock %}
```

VIEW DISEASE DETAILS PAGE CODING

```
{% extends 'index.html' %}

{% load static %}

{% block body %}

<div class="container-fluid" style="width:90%;margin-top:8%">
```

```
        <div class="container-fluid">
            <h1 align="center" style="font-weight:bold;font-family : 'Monotype Corsiva' ; color : #E6120E ;margin-top:4%">View Disease</h1>
        </div><hr>
<table id="example" class="display" style="width:100%">
        <thead>
                <tr>
                        <th>#</th>
                        <th>Disease Name</th>
                        <th>Type</th>
                        <th>Symptom of Disease</th>
                        <th>Action</th>
                </tr>
        </thead>
<tbody>
          {% for i in dis %}
                <tr>
                        <td>{{forloop.counter}}</td>
                        <td>{{i.name}}</td>
                        <td>{{i.type.name}}</td>
                        <td>{{i.symptom}}</td>
                        <td style="width:150px">
```

```
<a href="{% url 'edit_disease' i.id %}" ><button class="btn btn-primary"><i class="fa fa-edit"></i></button></a>

<a href="{% url 'delete_disease' i.id %}" ><button class="btn btn-danger" onclick="return confirm('Are you sure?')"><i class="fa fa-trash-o"></i></button></a></td>

        </tr>

    {% endfor %}

</tbody>

</table>

</div>

{% endblock %}
```

VIEW ALL DOCTORS PAGE CODING

```
{% extends 'index.html' %}

{% load static %}

{% block body %}

<div class="container-fluid" style="width:90%;margin-top:8%">

            <div class="container-fluid">

            <h1 align="center" style="font-weight:bold;font-family : 'Monotype Corsiva' ; color : #E6120E ;margin-top:4%">View Doctor</h1>

            </div><hr>

              <table id="example" class="display" style="width:100%">

              <thead>

              <tr>
```

```
                    <th>#</th>
                    <th>Full Name</th>
                    <th>Image</th>
                    <th>Email</th>
                    <th>Contact</th>
                    <th>Address</th>
                    <th>Category</th>
                    <th>Action</th>
                </tr>
            </thead>
<tbody>
        {% for i in doc %}
                <tr>
                    <td>{{forloop.counter}}</td>
                    <td>{{i.user.first_name}} {{i.user.last_name}}</td>
                    <td><img src="{{i.image.url}}" style="width:80px;height:80px"></td>
                    <td>{{i.user.email}}</td>
                    <td>{{i.contact}}</td>
                    <td>{{i.address}}</td>
                    <td>{{i.category}}</td>
                    <td style="width:150px">
```

```
                        <a href="{% url 'edit_doctor' i.id %}" ><button class="btn btn-
primary"><i class="fa fa-edit"></i></button></a>

                        <a href="{% url 'delete_doctor' i.id %}" ><button class="btn btn-danger"
onclick="return confirm('Are you sure?')"><i class="fa fa-trash-o"></i></button></a></td>

                </tr>

        {% endfor %}

</tbody>

</table>

</div>

{% endblock %}
```

CHANGE PASSWORD PAGE CODING

```
{% extends 'usernavigation.html' %}

{% block body %}

{% load static %}

<style>

.mybtnone:hover{background-color : #800000;

color : #429E00 ; font-weight : bold

}

</style>

<div class="container">
```

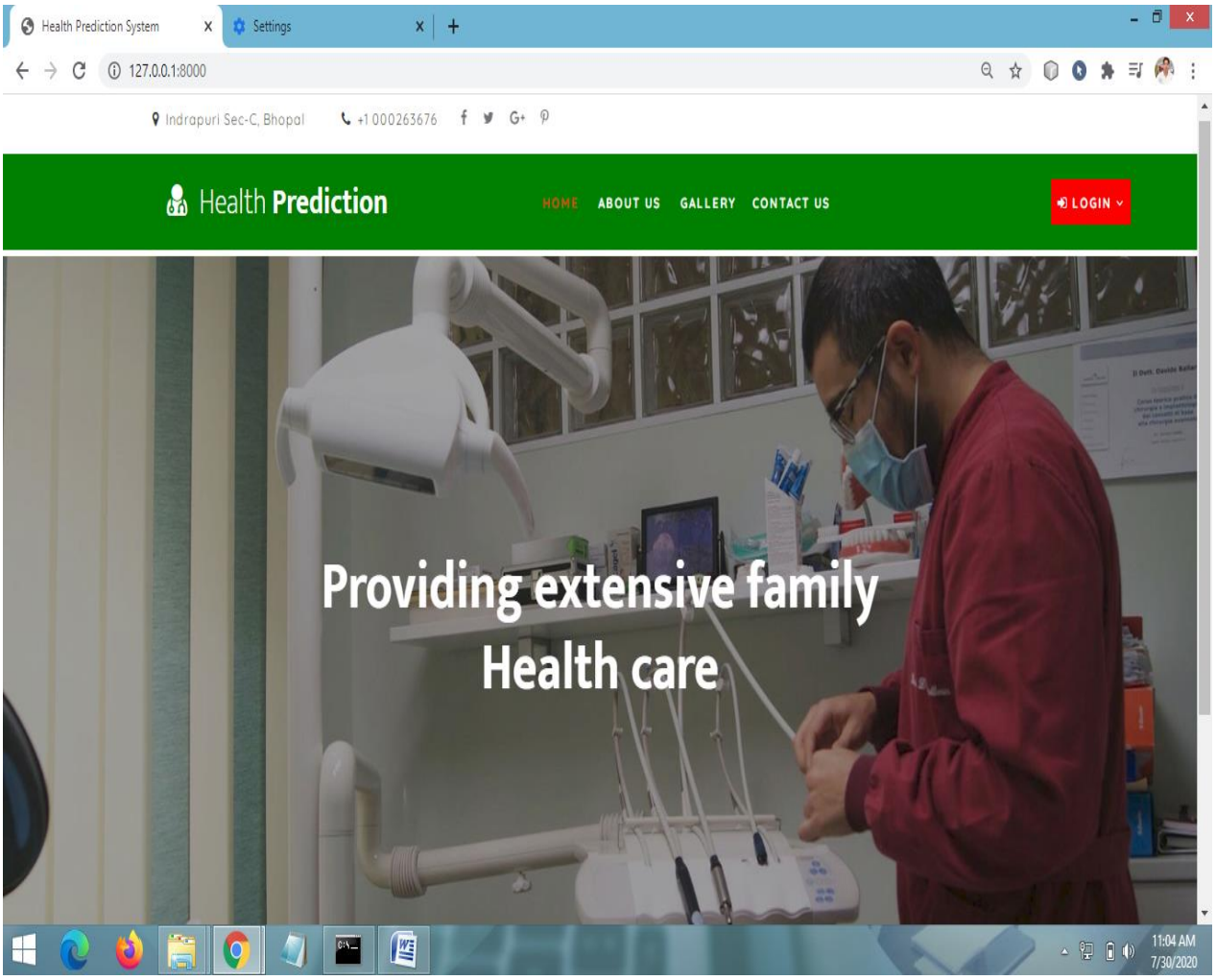
```
<h2  style = "font-weight:bold;font-family : 'Monotype Corsiva' ; color : #E6120E ;margin-top:7%" align="center">Change Password</h2><hr>

<div class="container-fluid" style="">

<div class="row">

<div class="col-md-6">

<form method="post" action="">

{% csrf_token %}

<div class="form-group">

<label >Old Password</label>

<input type="password" class="form-control"  aria-describedby="emailHelp" name="pwd3">

</div>

<div class="form-group">

<label >New Password</label>

<input type="password" class="form-control"  aria-describedby="emailHelp" name="pwd1">

</div>

<div class="form-group">

<label for="exampleInputPassword1">Confirm Password</label>

<input type="password" class="form-control" id="exampleInputPassword1"  name="pwd2">

</div>

<button type="submit" class="btn btn-primary mybtnone">Submit</button>

</form>

</div>

<div class="col-md-6">
```

```
<img src="{% static 'images/notes2.jpg' %}" style="width:90%;height:400px">

</div>

</div>

</div>

</div>

{% ifequal error "yes" %}

<script>

    alert('Password Changed.....');

    window.location=('{% url 'logout' %}')

</script>

{% endifequal %}

{% ifequal error "not" %}

<script>

    alert('New Password and Confirm Password are not match');

</script>

{% endifequal %}

{% endblock %}
```

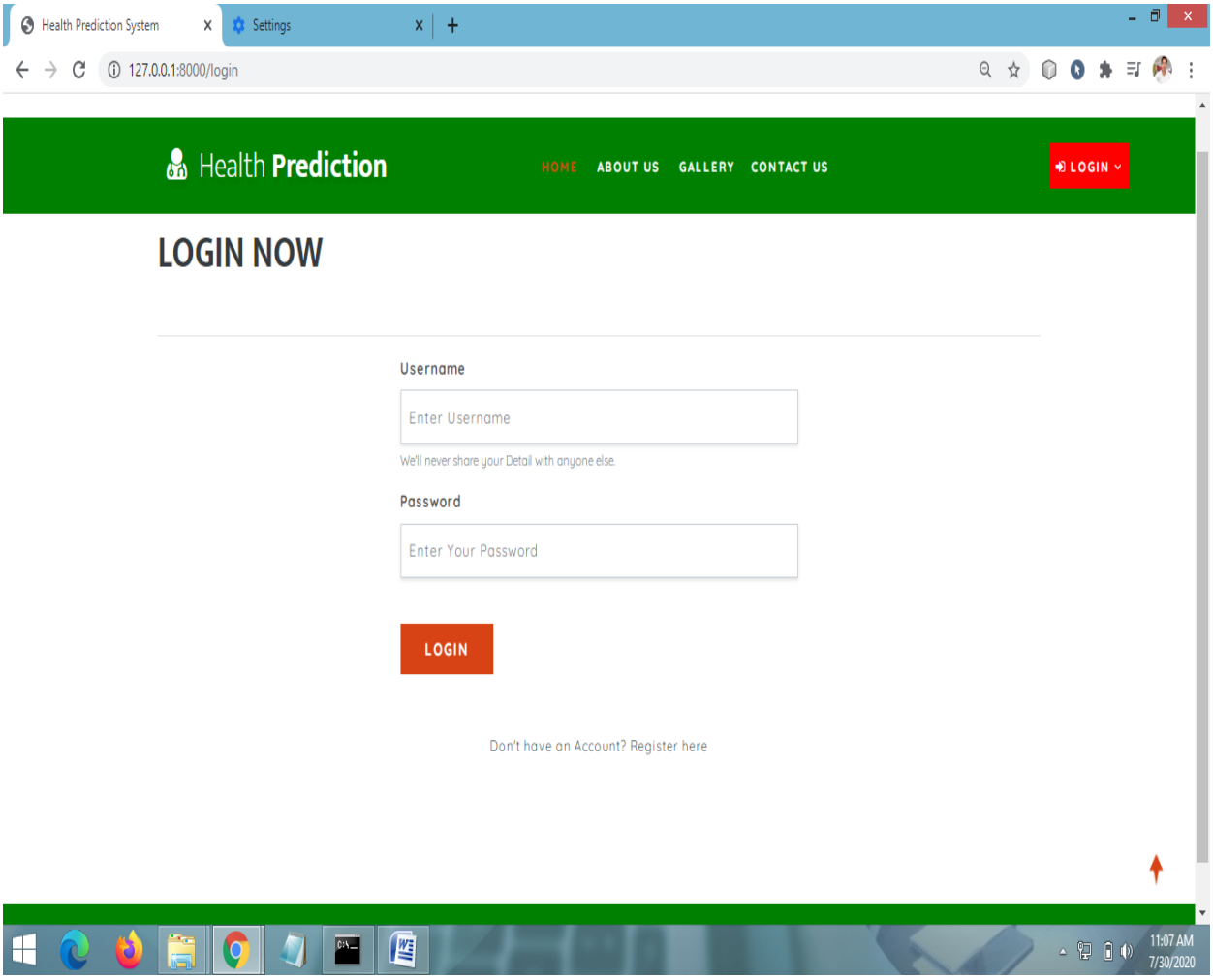
6. SCREENSHOTS

SCREENSHOTS

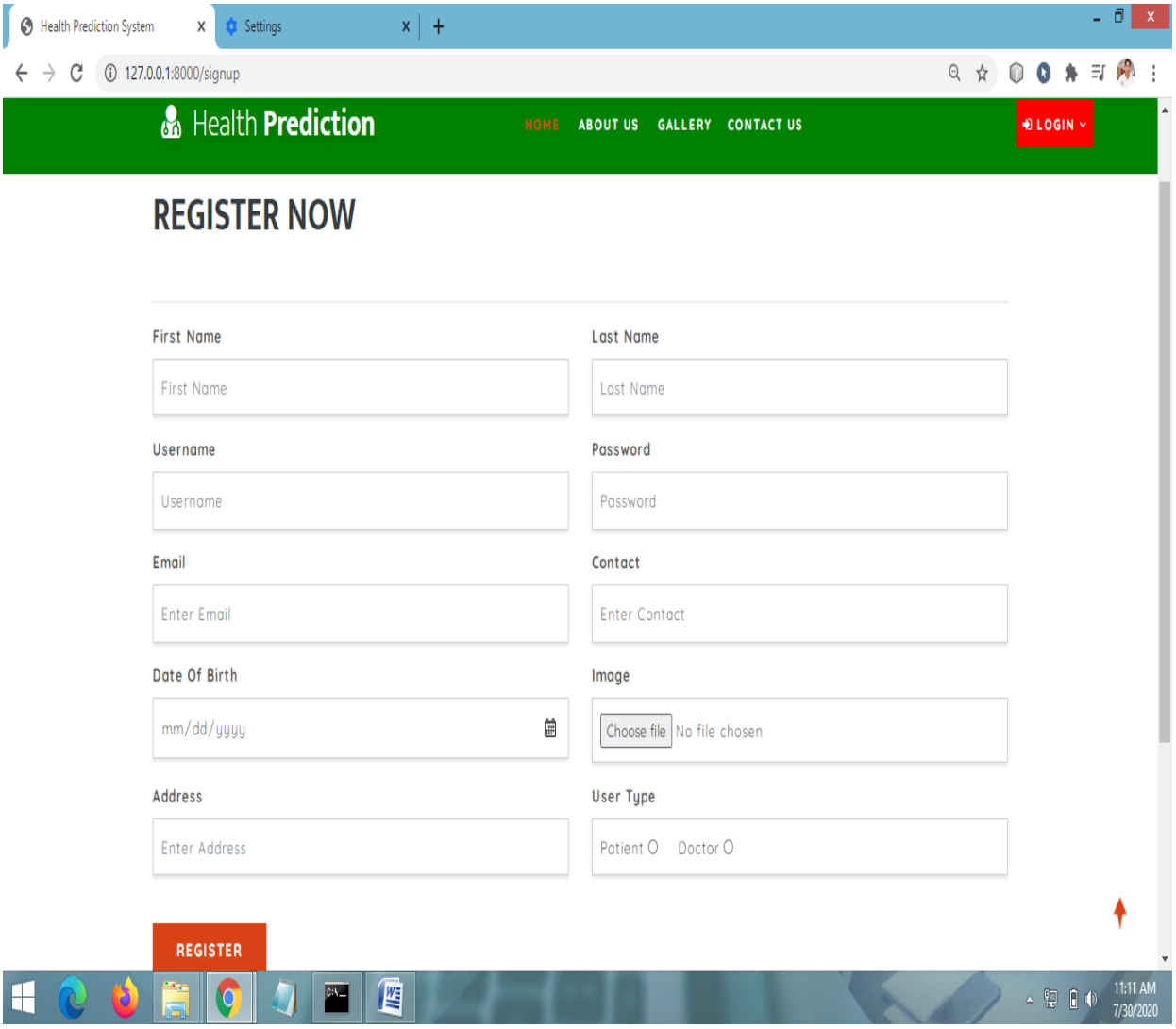
1. HOME PAGE



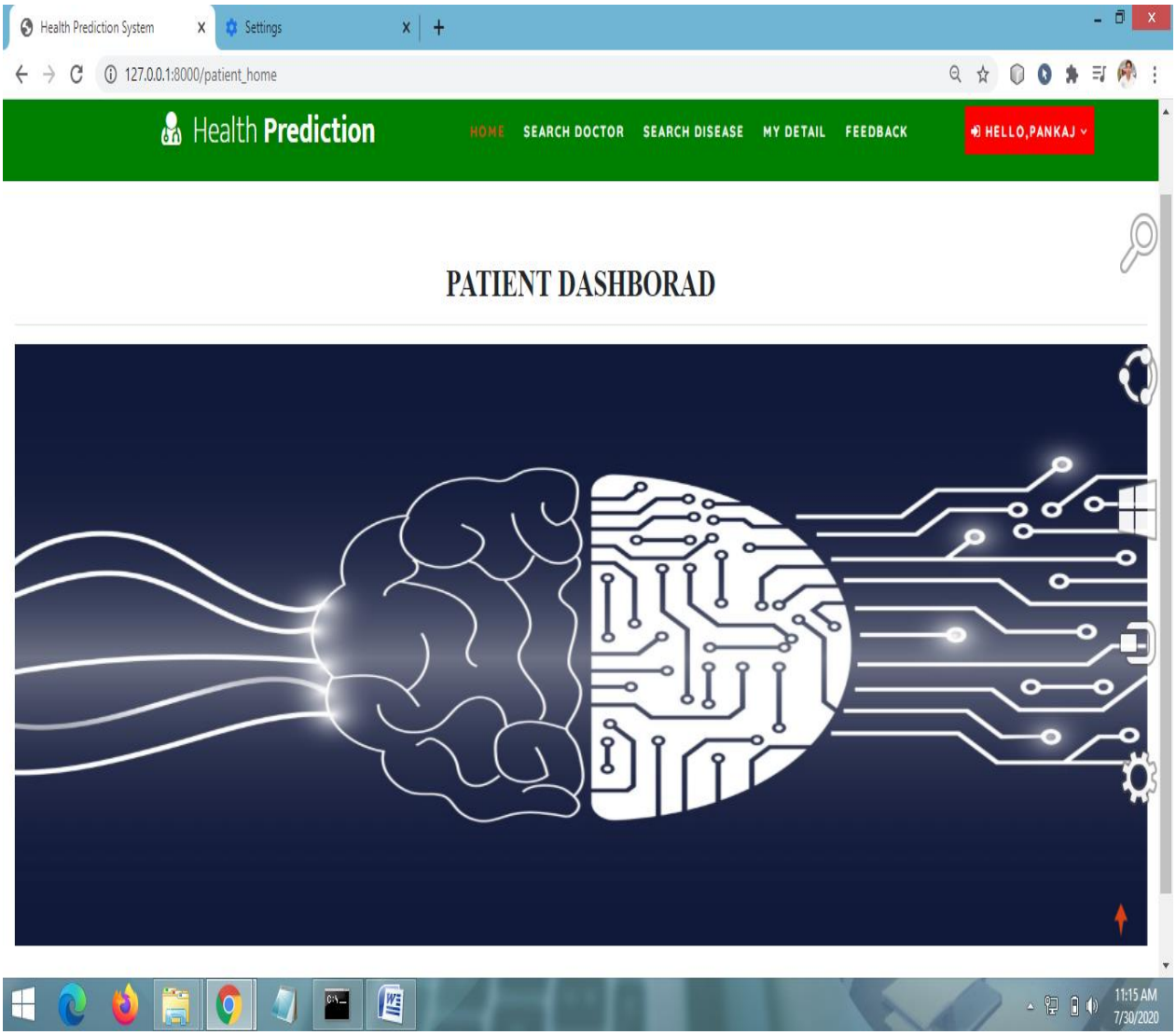
2. USER (PATIENT) LOGIN PAGE



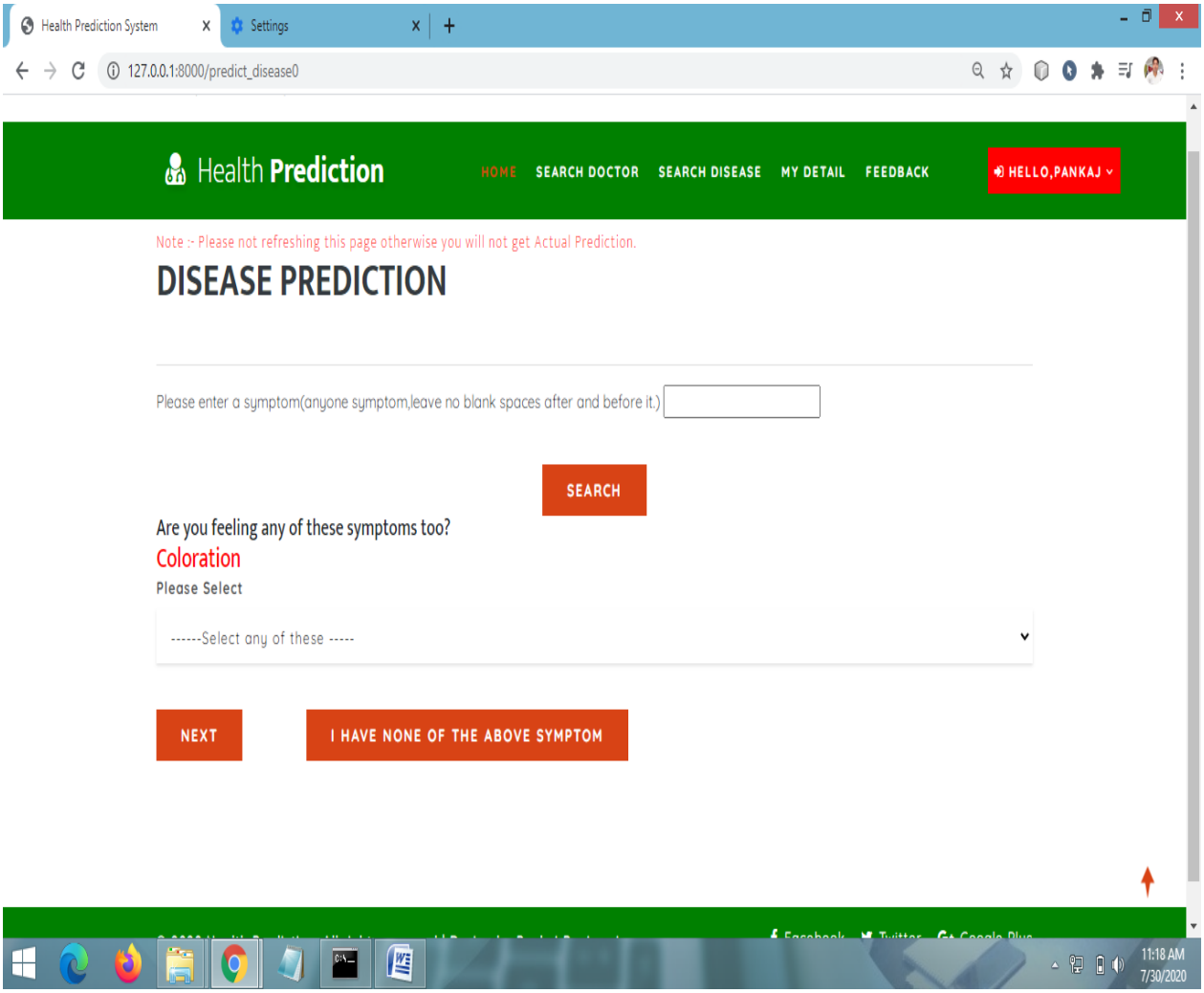
3. **SIGNUP PAGE**



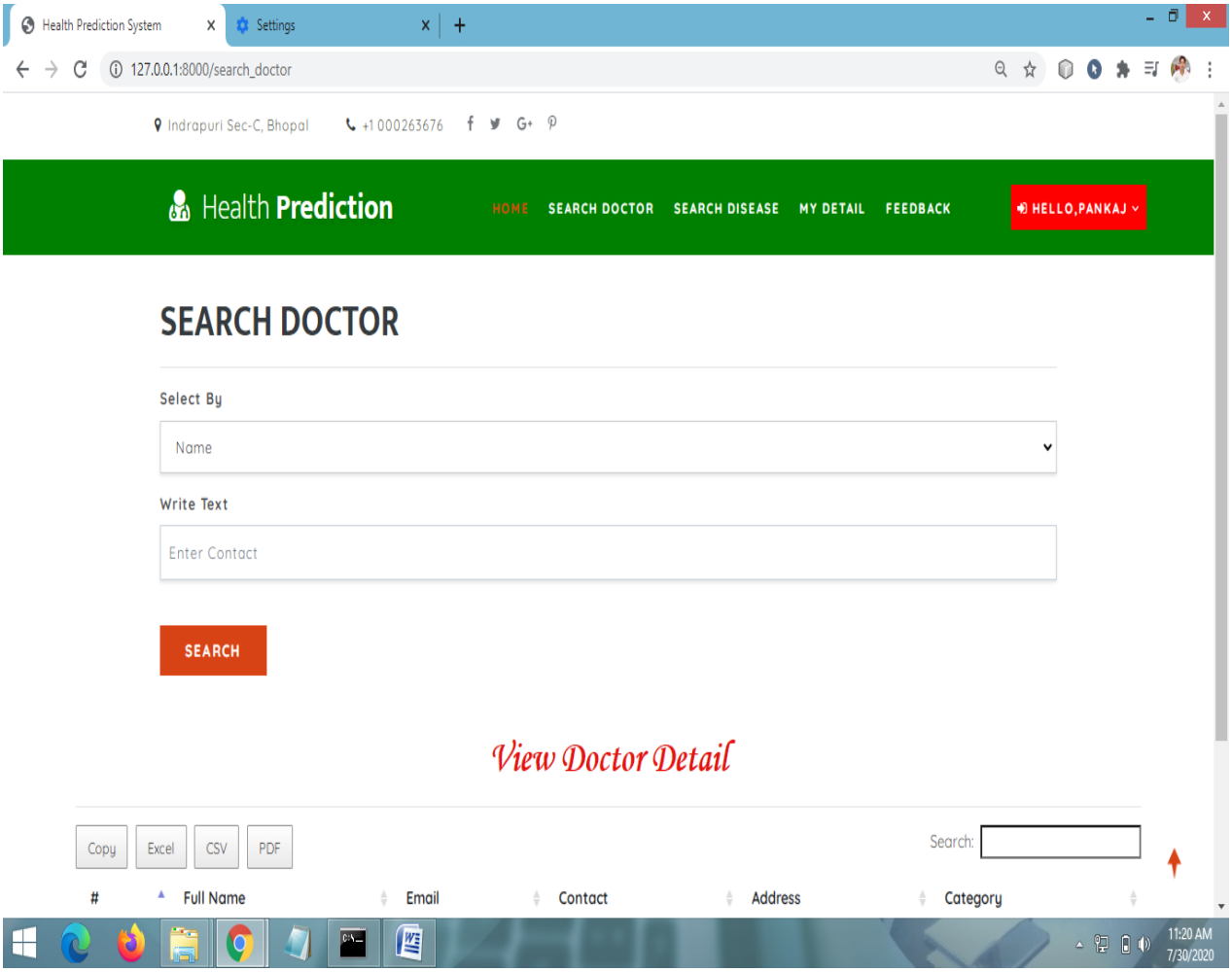
4. PATIENT HOME PAGE



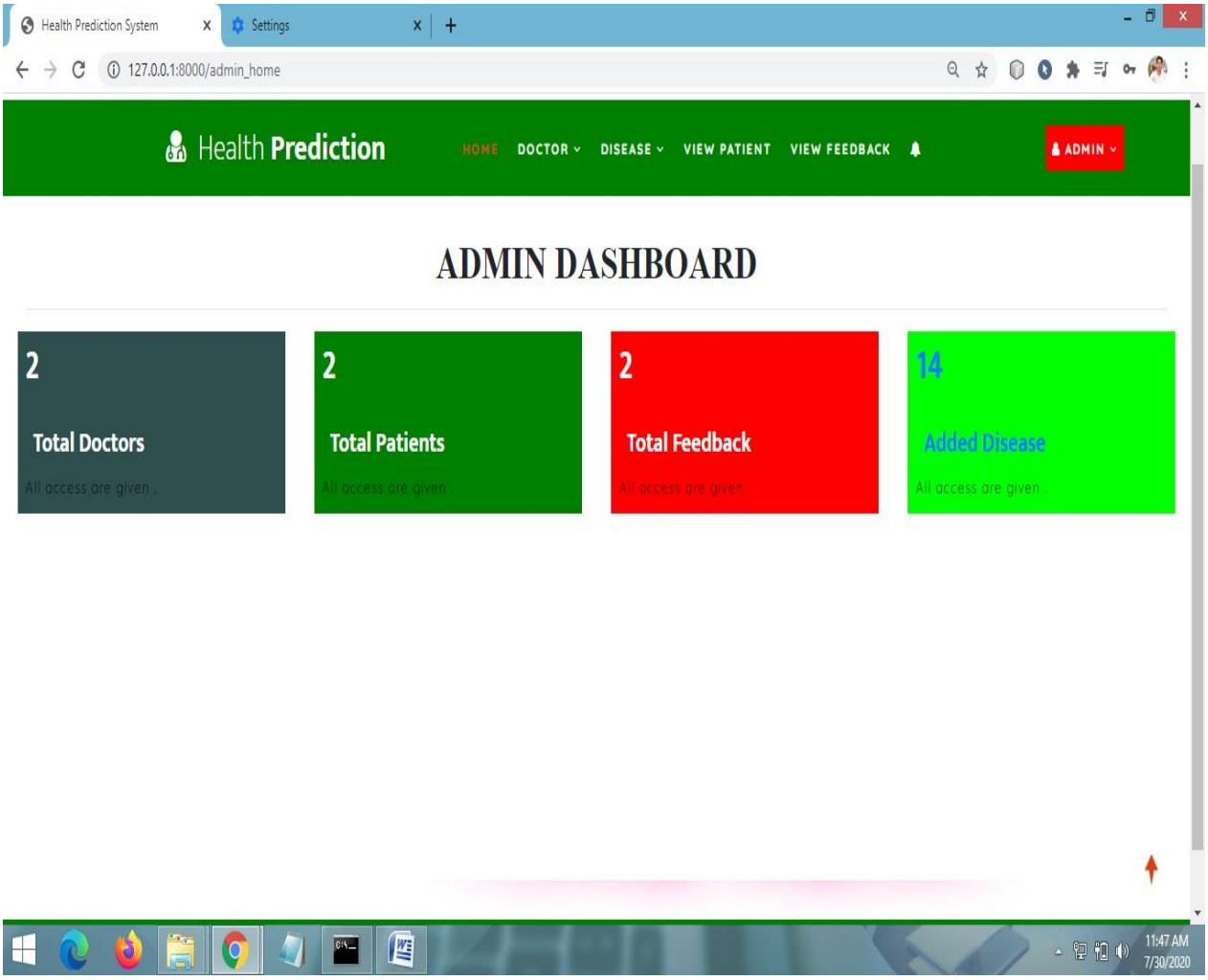
5. VIEW DISEASE PREDICTION RESULT PAGE



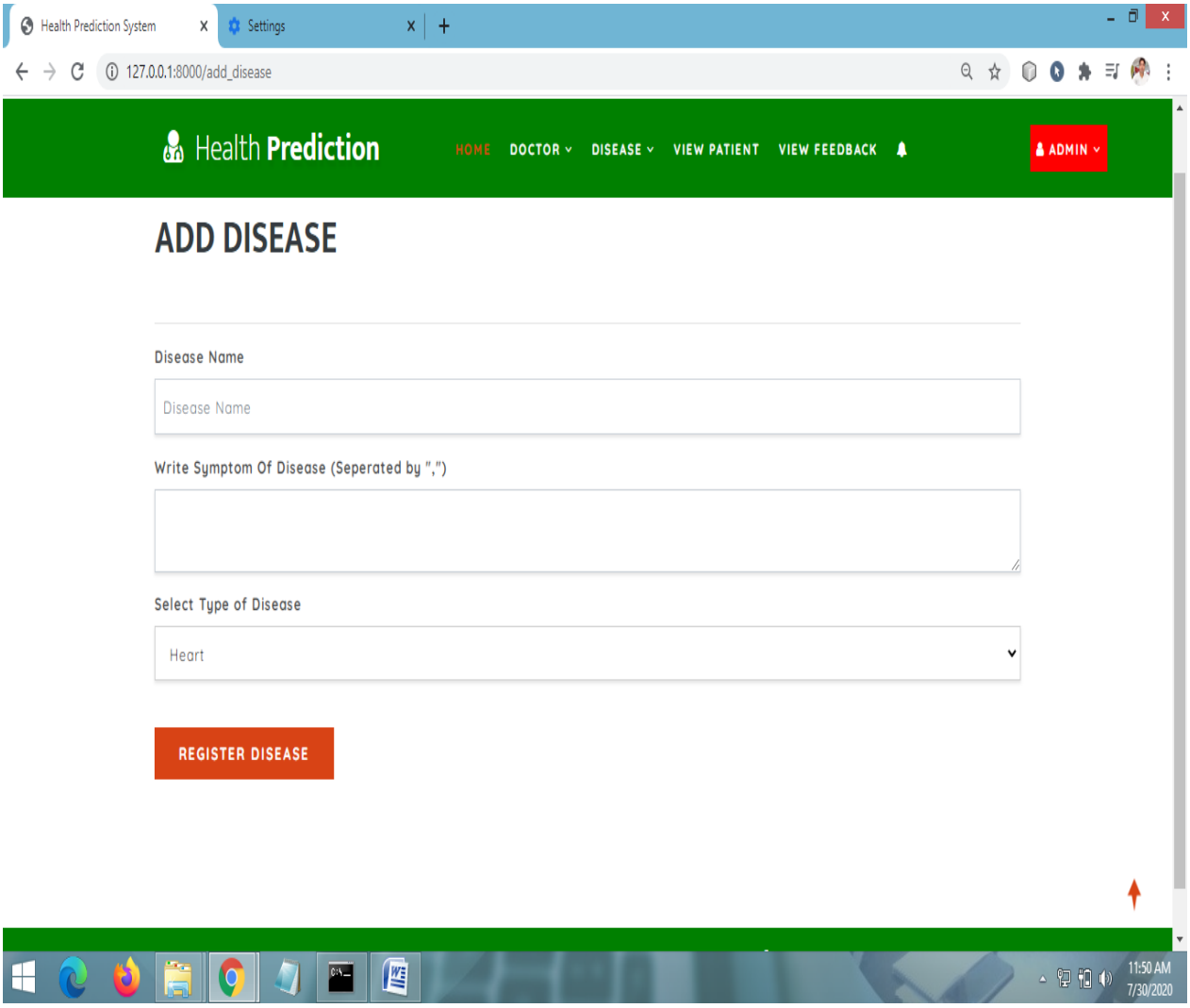
6. SEARCH DOCTOR PAGE



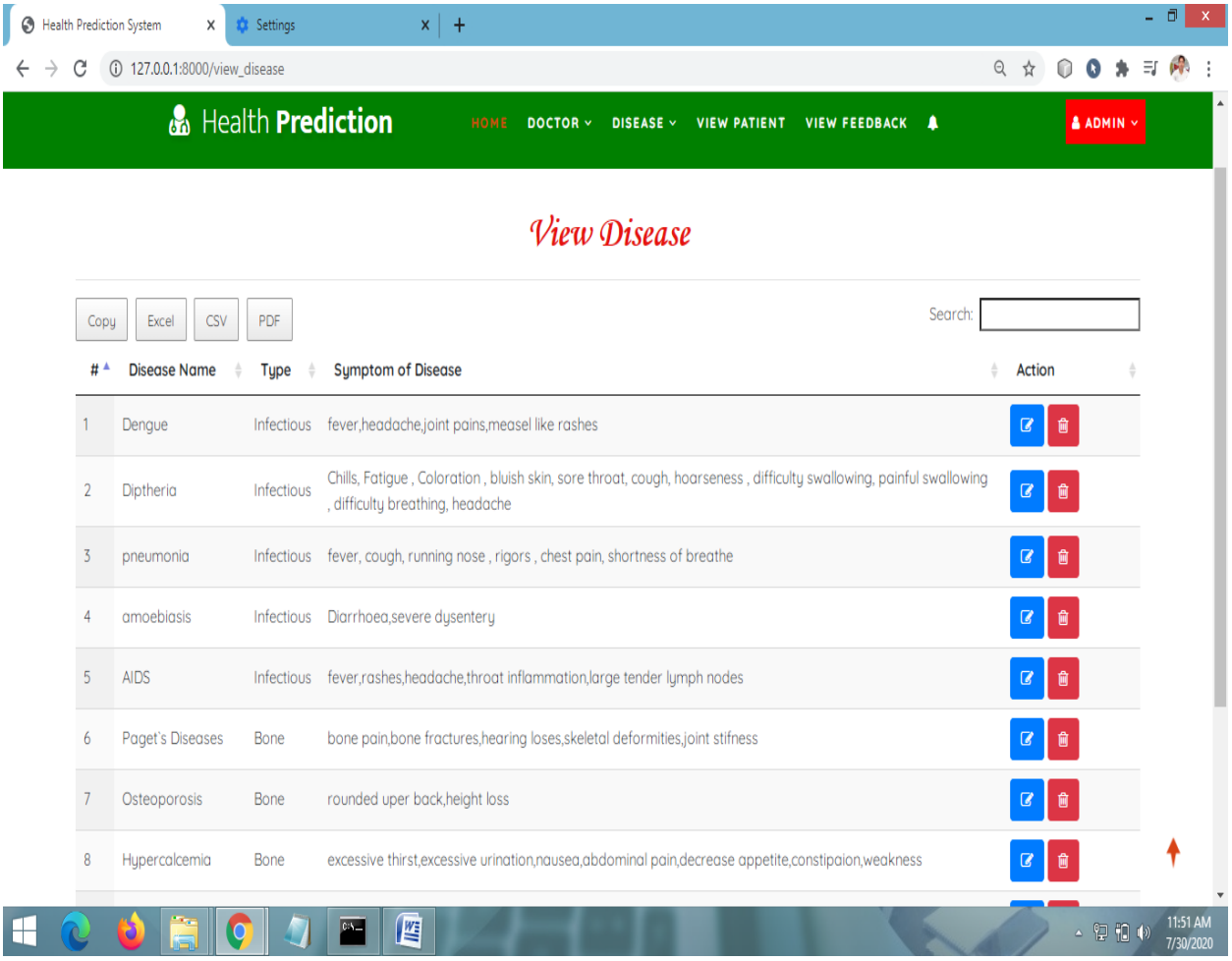
6. ADMIN HOME PAGE



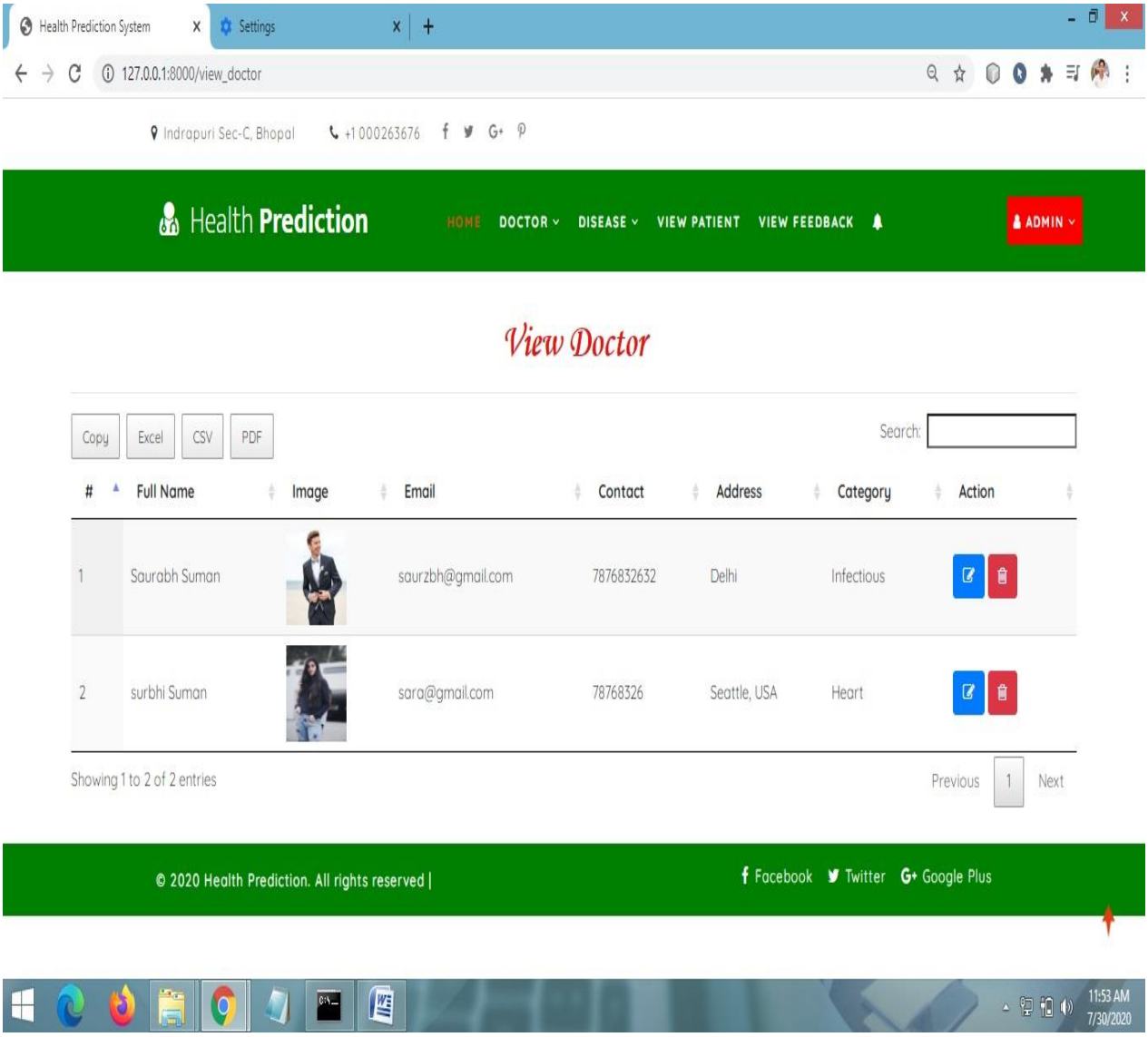
7. ADD DISEASE INFORMATION PAGE



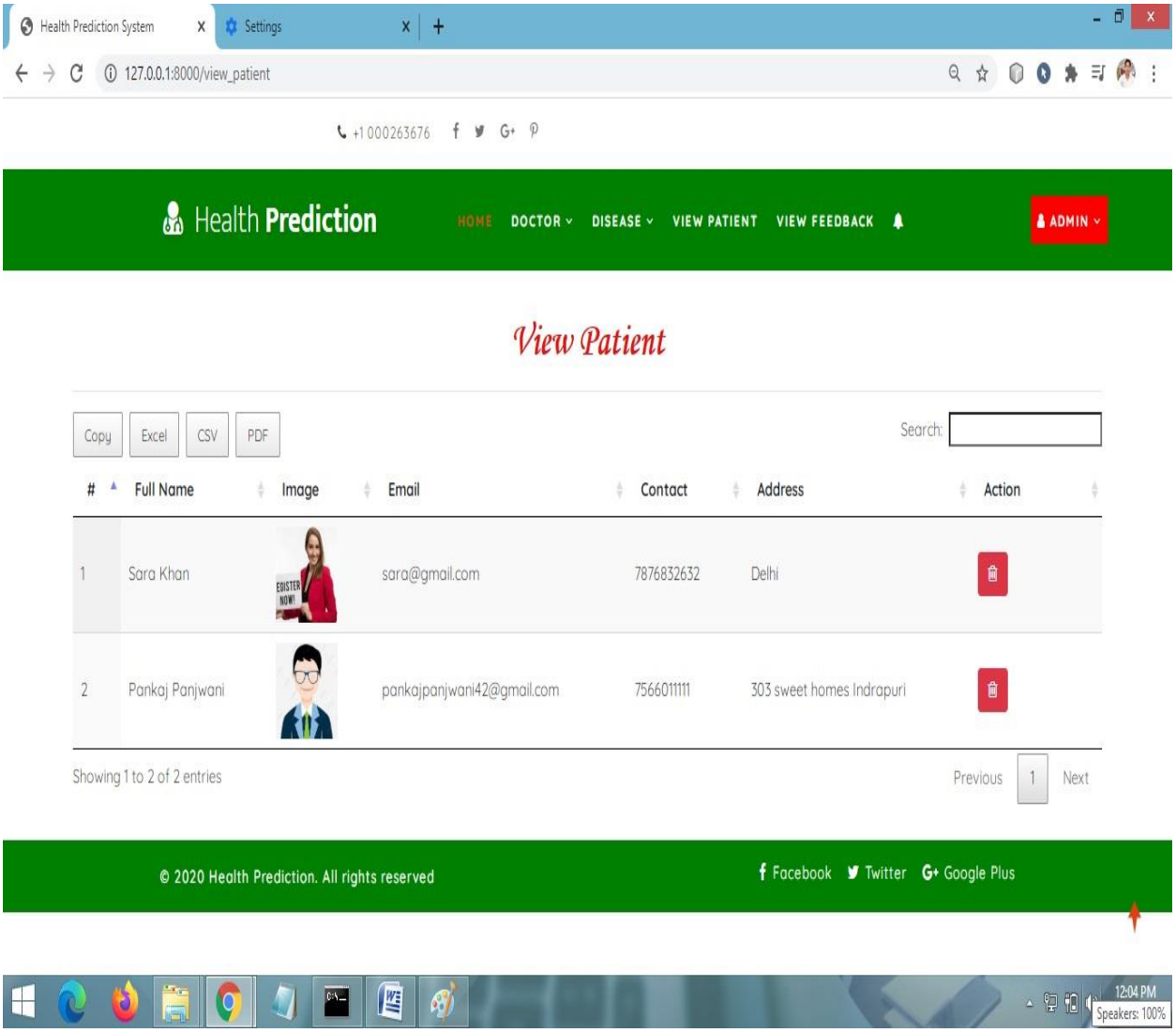
8. VIEW DISEASE INFORMATION PAGE


Health Prediction System
Settings
127.0.0.1:8000/view_disease
Health Prediction
HOME
DOCTOR
DISEASE
VIEW PATIENT
VIEW FEEDBACK
ADMIN
View Disease
Copy
Excel
CSV
PDF
Search:
Disease Name Type Symptom of Disease Action
1 Dengue Infectious fever,headache,joint pains,measel like rashes
2 Diptheria Infectious Chills, Fatigue , Coloration , bluish skin, sore throat, cough, hoarseness , difficulty swallowing, painful swallowing , difficulty breathing, headache
3 pneumonia Infectious fever, cough, running nose , rigors , chest pain, shortness of breathe
4 amoebiasis Infectious Diarrhoea,severe dysentery
5 AIDS Infectious fever,rashes,headache,throat inflammation,large tender lymph nodes
6 Paget's Diseases Bone bone pain,bone fractures,hearing loses,skeletal deformities,joint stifness
7 Osteoporosis Bone rounded uper back,height loss
8 Hypercalcemia Bone excessive thirst,excessive urination,nausea,abdominal pain,decrease appetite,constipaion,weakness
11:51 AM
7/30/2020

9. VIEW ALL DOCTORS PAGE



10. VIEW PATIENTS DETAIL



7. SYSTEM TESTING

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.1 TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the

combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.2 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

1. White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

2. Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- □ All field entries must work properly.
- □ Pages must be activated from the identified link.
- □ The entry screen, messages and responses must not be delayed.

Features to be tested

- □ Verify that the entries are of the correct format
- □ No duplicate entries should be allowed
- □ All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one stepup – software applications at the company level – interact without error.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results

All the test cases mentioned above passed successfully. No defects encountered.

8. CONCLUSION

8.1 CONCLUSION

At the end of this proposal we want to remember that this is fully unique system and we believe that it will helpful for us all as well as any Hospital Business can add this with their existing features. Hope this system will be very demandable in coming future.

The software takes care of all the requirements of an average hospital and is capable to provide easy and effective storage of information related to patients that come up to the hospital.

It generates test reports; provide prescription details including various tests, diet advice, and medicines prescribed to patient and doctor. It also provides injection details and billing facility on the basis of patient's status whether it is an indoor or outdoor patient.

The system also provides the facility of backup as per the requirement.

Mainly references don't need for this proposal because we make this from our own logical thinking/concept. But our Teacher/Instructor gives some directions to do this successful project proposal.

8.2 SCOPE FOR FUTURE ENHANCEMENT

It might have happened so many times that you or someone of yours need doctor's help immediately, but they are not available due to some reason. The Health Prediction system is an end user support and online consultation project.

The system allows user to share their symptoms and issues. Then the system processes user's symptoms to check for various illnesses that could be associated with it.

In doctor's module when a doctor login to the system, doctor can view his patient's details and the report of that patient. Doctor can view details about the patient and what patient searched for according to their prediction. Doctor can view his personal details. Admin can add new disease details by specifying the type and symptoms of the disease into the database.

This feasibility checks whether the system can be developed with the available funds. The Smart health prediction does not require enormous amount of money to be developed. This can be done

economically if planned judicially, so it is economically feasible. The cost of project depends upon the number of required.

This is the document of the Project proposal for developing a health care system. This system is fully updated and we had provide some new things in it. The project aimed to build a fully functional system in order to achieve the efficiency in faster heath treatment and online consultation system. The overall mission of system development is to make the primary treatment quickly and easily complete the Online Consultation System.

9. REFERENCES

References

1. *Kamal Acharya. School management system project report. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254873.34023165/v1*

2. *Kamal Acharya. A CASE STUDY OF CINEMA MANAGEMENT SYSTEM PROJECT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254873.30191075/v1*

3. *Kamal Acharya. A CASE STUDY ON ONLINE TICKET BOOKING SYSTEM PROJECT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254872.26972790/v1*

4. *Kamal Acharya. Web chatting application project report management system. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254871.18588592/v1*

5. *Kamal Acharya. RETAIL STORE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254871.14590154/v1*

6. *Kamal Acharya. SUPERMARKET MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252491.19145062/v1*

7. *Kamal Acharya. SOCIAL MEDIA MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252491.11210579/v1*

8. *Kamal Acharya. Online music portal management system project report. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252488.89734698/v1*

9. *Kamal Acharya. COLLEGE BUS MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 31, 2024. DOI: https://doi.org/10.22541/au.172245277.70798942/v1*

10. *Kamal Acharya. AUTOMOBILE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 31, 2024. DOI: https://doi.org/10.22541/au.172245276.67982593/v1*

11. *Kamal Acharya. Ludo management system project report. Authorea. July 31, 2024. DOI: https://doi.org/10.22541/au.172243999.98091616/v1*

12. *Kamal Acharya. Literature online quiz system project report. Authorea. July 31, 2024 DOI: https://doi.org/10.22541/au.172243825.53562953/v1*

13. *Kamal Acharya. Avoid waste management system project. Authorea. July 29, 2024. DOI: https://doi.org/10.22541/au.172228528.85022205/v1*

14. *Kamal Acharya. CHAT APPLICATION THROUGH CLIENT SERVER MANAGEMENT SYSTEM PROJECT. Authorea. July 29, 2024. DOI: https://doi.org/10.22541/au.172228527.74316529/v1*

15. *Acharya, Kamal, Online Job Portal Management System (May 5, 2024). Available at SSRN: https://ssrn.com/abstract=4817534* or *http://dx.doi.org/10.2139/ssrn.4817534*

16. *Acharya, Kamal, Employee leave management system. (May 7, 2024). Available at SSRN: https://ssrn.com/abstract=4819626* or *http://dx.doi.org/10.2139/ssrn.4819626*

17. *Acharya, Kamal, Online electricity billing project report. (May 7, 2024). Available at SSRN: https://ssrn.com/abstract=4819630* or *http://dx.doi.org/10.2139/ssrn.4819630*

18. Acharya, Kamal, POLICY MANAGEMENT SYSTEM PROJECT REPORT. (December 10, 2023). Available at SSRN: https://ssrn.com/abstract=4831694 or http://dx.doi.org/10.2139/ssrn.4831694

19. Acharya, Kamal, Software testing for project report. (May 16, 2023). Available at SSRN: https://ssrn.com/abstract=4831028 or http://dx.doi.org/10.2139/ssrn.4831028

20. Acharya, Kamal, ONLINE CRIME REPORTING SYSTEM PROJECT. (August 10, 2022). Available at SSRN: https://ssrn.com/abstract=4831015 or http://dx.doi.org/10.2139/ssrn.4831015

21. Acharya, Kamal, Burger ordering system project report. (October 10, 2022). Available at SSRN: https://ssrn.com/abstract=4832704 or http://dx.doi.org/10.2139/ssrn.4832704

23. Acharya, Kamal, Teachers Record Management System Project Report (December 10, 2023). Available at SSRN: https://ssrn.com/abstract=4833821 or http://dx.doi.org/10.2139/ssrn.4833821

23. Acharya, Kamal, Dairy Management System Project Report (December 20, 2020). Available at SSRN: https://ssrn.com/abstract=4835231 or http://dx.doi.org/10.2139/ssrn.4835231

24. Acharya, Kamal, Electrical Shop Management System Project (December 10, 2019). Available at SSRN: https://ssrn.com/abstract=4835238 or http://dx.doi.org/10.2139/ssrn.4835238

25. Acharya, Kamal, Online book store management system project report. (Febuary 10, 2020). Available at SSRN: https://ssrn.com/abstract=4835277 or http://dx.doi.org/10.2139/ssrn.4835277

26. Acharya, Kamal, Paint shop management system project report. (January 10, 2019). Available at SSRN: https://ssrn.com/abstract=4835441 or http://dx.doi.org/10.2139/ssrn.4835441

27. Acharya, Kamal, Supermarket billing system project report. (August 10, 2021). Available at SSRN: https://ssrn.com/abstract=4835474 or http://dx.doi.org/10.2139/ssrn.4835474

28. Acharya, Kamal, Online taxi booking system project report. (March 10, 2022). Available at SSRN: https://ssrn.com/abstract=4837729 or http://dx.doi.org/10.2139/ssrn.4837729

29. Acharya, Kamal, Online car servicing system project report. (March 10, 2023). Available at SSRN: https://ssrn.com/abstract=4837832 or http://dx.doi.org/10.2139/ssrn.4837832

30. Acharya, Kamal, School management system project report. (July 10, 2021). Available at SSRN: https://ssrn.com/abstract=4837837 or http://dx.doi.org/10.2139/ssrn.4837837

31. Acharya, Kamal, Furniture Showroom Management System Project Report (March 21, 2021). Available at SSRN: https://ssrn.com/abstract=4839422 or http://dx.doi.org/10.2139/ssrn.4839422

32. Acharya, Kamal, Online Vehicle Rental System Project Report (March 21, 2019). Available at SSRN: https://ssrn.com/abstract=4839429 or http://dx.doi.org/10.2139/ssrn.4839429

33. Acharya, Kamal, Fruit Shop Management System Project Report (August 10, 2023). Available at SSRN: https://ssrn.com/abstract=4841048 or http://dx.doi.org/10.2139/ssrn.4841048

34. Acharya, Kamal, Hall Booking Management System Project Report (December 21, 2023). Available at SSRN: https://ssrn.com/abstract=4841055 or http://dx.doi.org/10.2139/ssrn.4841055

35. Acharya, Kamal, Lundry Management System Project Report (October 21, 2023). Available at SSRN: https://ssrn.com/abstract=4841059 or http://dx.doi.org/10.2139/ssrn.4841059

36. *Acharya, Kamal, A CASE STUDY OF CINEMA MANAGEMENT SYSTEM PROJECT (September 25, 2023). Available at SSRN: https://ssrn.com/abstract=4841209* or *http://dx.doi.org/10.2139/ssrn.4841209*

37. *Acharya, Kamal, A CASE STUDY ON ONLINE TICKET BOOKING SYSTEM PROJECT (May 25, 2024). Available at SSRN: https://ssrn.com/abstract=4841210* or *http://dx.doi.org/10.2139/ssrn.4841210*

38. *Acharya, Kamal, ONLINE DATING MANAGEMENT SYSTEM PROJECT REPORT. (April 25, 2023). Available at SSRN: https://ssrn.com/abstract=4842066 or http://dx.doi.org/10.2139/ssrn.4842066*

39. *Acharya, Kamal, ONLINE RESUME BUILDER MANAGEMENT SYSTEM PROJECT REPORT. (April 25, 2021). Available at SSRN: https://ssrn.com/abstract=4842071 or http://dx.doi.org/10.2139/ssrn.4842071*

40. *Acharya, Kamal, TOLL TEX MANAGEMENT SYSTEM PROJECT REPORT (August 21, 2023). Available at SSRN: https://ssrn.com/abstract=4842082 or http://dx.doi.org/10.2139/ssrn.4842082*

41. *Acharya, Kamal, Chat Application Through Client Server Management System Project Report (June 25, 2023). Available at SSRN: https://ssrn.com/abstract=4842761 or http://dx.doi.org/10.2139/ssrn.4842761*

42. *Acharya, Kamal, Web Chatting Application Management System Project Report (April 25, 2022). Available at SSRN: https://ssrn.com/abstract=4842771 or http://dx.doi.org/10.2139/ssrn.4842771*

43. *Acharya, Kamal, Automobile management system project report (May 25, 2022). Available at SSRN: https://ssrn.com/abstract=4846917 or http://dx.doi.org/10.2139/ssrn.4846917*

44. *Acharya, Kamal, College bus management system project report (April 25, 2023). Available at SSRN: https://ssrn.com/abstract=4846920 or http://dx.doi.org/10.2139/ssrn.4846920*

45. *Acharya, Kamal, Courier management system project report (May 25, 2023). Available at SSRN: https://ssrn.com/abstract=4846922 or http://dx.doi.org/10.2139/ssrn.4846922*

46. *Acharya, Kamal, Event management system project report (April 25, 2021). Available at SSRN: https://ssrn.com/abstract=4846927 or http://dx.doi.org/10.2139/ssrn.4846927*

47. *Acharya, Kamal, Library management system project report II (May 25, 2020). Available at SSRN: https://ssrn.com/abstract=4848857 or http://dx.doi.org/10.2139/ssrn.4848857*